Issues in the Incorporation of Security Services into a Protocol Reference Model*†

Mahesh V. Tripunitara and Eugene H. Spafford
COAST Laboratory
Purdue University
West Lafayette, IN 47907-1398
{*tripunit,spaf*}*@cs.purdue.edu*

COAST TR-98/03

Abstract

A Protocol Reference Model is an abstraction of the communication subsystem of a system. Thus, it is appropriate to focus on the protocol reference model when examining the issue of secure communications.

In this paper, we discuss some issues in incorporating security services into a protocol reference model. The security services considered are authentication, confidentiality, integrity and access control. We adopt a functional definition for a protocol reference model in terms of the communication services it provides at various layers. We then present two perspectives towards reasoning about the incorporation of security services into a protocol reference model: a perspective that centers on the security requirements, and another that centers on the communication services already present in the protocol reference model.

Existing work focuses on the first approach. We focus on the second approach, that is, on the issue of how well a security service slated for incorporation meshes in with the existing communication services provided at a layer in the protocol reference model. By considering communication services at a lower level of abstraction, in terms of the controls and the associated variables that are used to realize it, we present a criterion on which to base the incorporation of security services into a layer. We assume that each control is modeled as a Mealy machine and that each state is labeled with a particular set of values for the variables. We then identify variables associated with the controls of the security services that capture their functionality, but isolate them from specific mechanisms to realize them. The criterion for incorporation of the security services into a layer requires that a security service be incorporated into a layer only if the variables associated with the security service are already present among the communication services offered at the layer. We then show the use of the criterion on two examples: the reference models behind the TCP/IP protocol suite and Asynchronous Transfer Mode (ATM).

The main contributions of this paper lie in the clarification of the different levels of abstraction to the problem, a treatment of the issues at a level of abstraction not found in existing work on the topic, which gives new insights, and in a concrete criterion on which to base the incorporation of security services into a protocol reference model.

1 Introduction

This paper addresses the issue of secure communications by focusing on an abstraction of the communication subsystem of a system. In this introductory section, we discuss the background and motivation

†Submitted to the Fifth ACM Conference on Computer and Communications Security.
*Portions of this work were supported by sponsors of the COAST Laboratory.

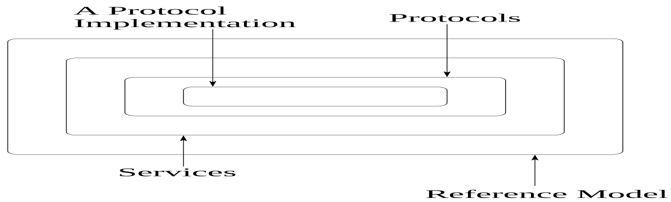

Figure 1: A protocol reference model, the services and protocols are three levels of abstraction.

relevant to the issues addressed in this paper. We also clarify the scope of our treatment. We introduce terms and phrases crucial to our discussions, while deferring full definitions to the relevant sections later in the paper.

1.1 Background

A *Protocol Reference Model* is an abstract description of the communication subsystem of a system and provides the basic framework for the interconnection and communication of two or more systems [29]. The Open Systems Interconnection (OSI) reference model [26, 29, 32], and the model that the TCP/IP protocol suite [18, 43, 44] is based on, are examples of such a model. A communication *service* is a functional component of a protocol reference model. The functionality within a protocol reference model is typically decomposed into *layers*. *Layering* is a method of dividing functionality (that is, the communication services) into separate domains, achieving a form of hierarchical modularity [56] (see figure 2). Part of the services offered by a layer are implemented as functionality within the layer, while the rest are derived from the services provided by lower layers [29]. Examples of communication services are ordered delivery of data, connection management and flow control (see section 2). An instance of the services provided at a layer is an *entity* [29]. The notion of an entity is similar to that of a process. A *protocol* is the set of rules and conventions by which two entities communicate [18, 24, 29, 41]. Thus, protocols are distributed algorithms that execute between peer instances of the same layer [56] and realize one or more of the communication services [32]. A *protocol stack* is a specific instance of the protocol reference model that places specific protocols at the various layers.

A protocol reference model, the communication services it is associated with and the protocols that realize those services can be thought of as three level of abstraction for a communication subsystem [29, 32]. Figure 1, which is taken from [29] clarifies this perspective.

In 1988, the ISO proposed a security architecture to the basic protocol reference model [26] in recommendation [27]. [27] proposes enhancing the services provided at the layers in the basic protocol reference model with services pertaining to security. Alternate recommendations have been made for the OSI Protocol Reference Model (see, for instance, [31, 46, 47, 50, 51]). Recommendations have also been developed for other networking technologies such as the TCP/IP protocol suite [18, 43, 44] (see for instance [2, 3, 4, 48, 49]) and Asynchronous Transfer Mode (ATM) [40] (see for instance, [16, 19, 42, 61]). The primary arguments against the ISO recommendation [27] are that it does not sufficiently substantiate its recommendations and is not definitive enough for implementors [22].

A *security service* is similar to a communication service, as discussed in the previous paragraph. But it is a service that partly or wholly fulfills a *security requirement.* An example of a security service is the *data confidentiality* service, which we define and discuss in section 3.2. Traditionally, security requirements have pertained to the *confidentiality* and *integrity* of data and the *availability* of resources to access such data [22]. We defer the definition of these terms and further discussion on security requirements from protocol reference models to section 3.1. Security services enhance the services already provided at a layer by fulfilling security requirements [31]. Thus, incorporation of security services into the layers of a protocol reference model begins with the identification of the security requirements from the protocol reference model. These security requirements are specified in, or gleaned from, a *security policy* [1] or in terms of the *security properties* on the communicated data and the consumed network resources. The ISO identified several security services for incorporation into the OSI protocol reference model in [27]. In this paper, we consider the same security services for incorporation into any protocol reference model. The issue of whether and how the security services fulfill the security requirements is beyond the scope of this paper, though we do touch on it in section 3.2.

A *security mechanism* is a realization of a security service, as a protocol, algorithm or heuristic. Examples of security mechanisms are the MD4 and MD5 message digest algorithms [52, 53], which could be used to realize the integrity service, the Data Encryption Standard [35], which could be used to realize the confidentiality service, and the Needham–Schroeder protocol [38], which could be used to realize the authentication service. We discuss security mechanisms further in section 3.3. Also, as we discuss in that section, security requirements, services and mechanisms are three levels of abstraction in the considerations for security as they pertain to a communication subsystem.

1.2 Motivation

Once the appropriate security services that need to be incorporated are identified, there remains the issue of how to incorporate these services into the various layers in a protocol reference model. Following are two approaches to this issue, each fulfilling a different goal.

- One approach seeks to answer the question of how well such incorporation serves towards fulfillment of the security requirements from the protocol reference model.
- Another approach seeks to answer the question of whether a security service in question is appropriate for incorporation into (or "meshes in well" with) a layer given the services already offered by the layer, and the assumptions that will have to be made about services offered by higher and lower layers in the protocol reference model for the reference model to provide the services we would like it to.

The first approach is due to the security services offered by a protocol reference model having to fulfill the user's[1] security requirements from the protocol reference model. It is desirable that incorporation of the security services be done in a way that the changes to be made to the existing services are minimized. In other words, security services should be added to as few layers in the protocol reference model as possible. The second approach is due to the security services enhancing the existing services offered by a layer and therefore needing to be "compatible with" those existing services. Thus, the first approach satisfies sufficient conditions for the incorporation of security services into a protocol reference model, with those conditions being set by the security requirements. The second approach satisfies necessary conditions, with those conditions being set by the protocol reference model and the communication services associated with it.

Existing work, such as [27, 31, 46, 47, 50, 51] in the context of the OSI model, [2, 3, 4, 48, 49] in the context of TCP/IP and [16, 19, 42, 61, 64] in the context of ATM provides instances for the first approach, in a way that changes to the protocol reference model are minimized. Formal models (such as those from [62, 63, 65, 66]) for specifying the security requirements from a protocol reference model are not used and therefore the papers only give informal justifications for how the suggested placement fulfills the security requirements. A characteristic of those papers is that there is no separation of security mechanisms from security services when considerations are made for placement.

1.3 Scope of This Paper

In this paper, we focus on the second approach and provide a criterion to determine where, in a protocol reference model, security services may be incorporated, and the assumptions that would need to be

[1] A *user* in this context is typically a network administrator.

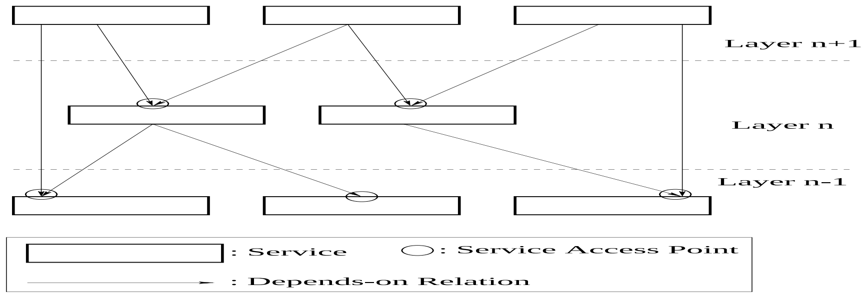

Figure 2: A Protocol Reference Model with Its Associated Services

made with each choice. Along the way, we discuss *protocol reference model*, *communication services* and *protocols*, and clarify the differences between *security requirements*, *services* and *mechanisms*. Then, we list several security services that we consider for incorporation, this list being drawn from the ISO recommendation [27]. We also point out that our consideration is limited to two communication models (see section 2.1). Then, as examples, we apply our criterion to TCP/IP and ATM and show into which layers in the relevant protocol reference models various security services can be incorporated.

2 Protocol Reference Model, Communication Services and Protocols

In this section, we define and discuss *protocol reference model*, *communication services*, *protocol* and the notion of *layering* as applied to a protocol reference model. These form the basis for the criterion we propose in section 4. We also describe the models for communication that are considered in this paper.

A *protocol reference model* is an abstract description of a communication subsystem in a system. Its primary purpose is to delineate the functionality contained within such a subsystem. A *communication service* is a function performed within a protocol reference model that aids a higher level entity in communication or in an improved quality of communication. Thus, the collection of all services within a protocol reference model would constitute a functional description of the protocol reference model. A modular and hierarchical division of the functionality with a protocol reference model is achieved by splitting the services across *layers*. Part of the services offered by a layer are implemented as functionality within the layer, while the rest are derived from the services provided by lower layers [29]. An instance of the services provided at a layer is an *entity* at that layer. Figure 2 is an example of a protocol reference model.

Our notion of a protocol reference model is similar to that of a *protocol graph* from [41]. Thus, as figure 2 shows, the nodes in the protocol reference model (which is a graph) are the services and an edge represents a *depends on* relation. A service that depends on another service uses the functionality provided by the latter to provide its own functionality. We have chosen to use directed edges (as opposed to undirected edges as used in [41]) for clarity. Thus, a protocol reference model is a topologically sorted graph in which the nodes represent services and the edges represent the "depends on" relation. The nodes that represent services that belong to a layer do not have edges between them.

Note that an edge from S_i^j to S_l^k where $i - l > 1$ is considered a layer violation in the OSI protocol reference model [26, 60], but we allow such edges so that services such as those from the TCP/IP protocol suite and ATM can be modeled within the framework proposed by figure 2.

An important component from figure 2 is that of the *service access point* (SAP). A SAP is the interface through which the functionality provided by the service can be accessed. A SAP is typically realized as an Application Programmer Interface (API) or as an exported method of an object.

Each service within a protocol reference model can be represented at a lower level of abstraction using a *control* with its associated *variables* [30, 31], as indicated in figure 3. The control is a formal model

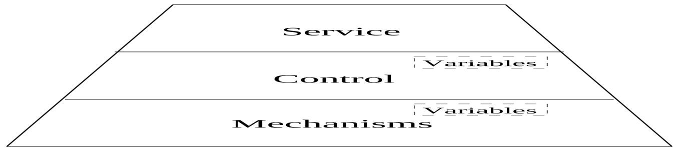

Figure 3: A Communication Service at three levels of abstraction

such as a finite state automaton [44] or petri net [33, 58]. OSI protocol development environments such as VOPS [30, 31] use extended finite state automata for the control. The Formal Description Techniques (FDT) Extended State Transition Language (ESTL or Estelle) [11, 28], which is an ISO standard, and the Specification and Description Language (SDL) [55], which is a CCITT standard, that are used for formally specifying protocol behaviour, also model the control as extended finite state automata. An extended finite state automaton is equivalent to a finite state automaton, but maintains auxiliary variables to reduce the number of states [54].

In this paper, we assume that the control is realized using a Mealy machine, which is a finite state automaton with output, where the output is determined by the state and the input [25]. Further, we assume that each state is labeled with the values of the variables that correspond to that state. For instance, the Mealy machine that models the control part for the connection management service that is realized in TCP [18, 44] can have its states labeled with the values for the two variables $SentValue$ and $ReceivedValue$, where each of those variables takes on values from the set $\{$ *syn*, *ack*, $syn + ack$, fin, *open*, *close*, *reset*, *anything* $\}$. Examples of communication services are Connection Management, Ordered Delivery, Flow and Congestion Control, Error Detection, Recovery and Control, Synchronization and Multiplexing [60].

The *mechanisms* in figure 3 are a realization of the controls as a *protocol* or algorithm. Note that figure 3 shows two sets of variables, one associated with the control and the other with the mechanisms. The distinction between the variables associated with the control part and those associated with the mechanisms is important, since they belong to two different levels of abstraction for the communication subsystem. Thus, the mechanisms could involve variables of their own, but such variables are not important to the issues discussed in this paper. We work at a level of abstraction above the mechanisms and focus only on the services.

A *protocol* is the set of rules and conventions by which two entities communicate [18, 24, 29, 41], and is a distributed algorithm that executes between peer instances (that is, entities) of the same layer [56]. A protocol is a realization of one or more of communication services [29, 32], and implements the controls associated with those services. A protocol receives Service Data Units (SDUs) through the service access point [29, 32] with a protocol at a higher layer. Thus, an SDU is the unit of input to a protocol. An instance of a protocol at the sender's end sends a *Protocol Data Unit* (PDU) to an instance of the protocol at the receiver's end. Thus, a PDU is a unit of communication between peer entities at a layer in two or more instances of a protocol reference model. The correspondence between SDUs and PDUs is dependent on the mechanisms used by the protocol in question. Examples of how this mapping is carried out at the sender are [26, 60]: segmentation, translation (from one format to another), encapsulation (includes addition of control information such as sequence numbers and error detection codes, and padding), and generation of PDUs that do not correspond to SDUs.

We wish to adopt and emphasize two principles of *layering* from the OSI model [26] in this paper:

- The variables, controls and mechanisms within a layer are not visible or accessible to another layer. The only way for one entity at a layer (or an application entity) to interact with an entity at a lower layer is through the predefined service access point.
- Along a path in the protocol reference model, suppose there is an edge from service P to service Q. To P, whatever follows starting at Q on the rest of the path is a protocol reference model. In other words, the service access point that Q provides represents the rest

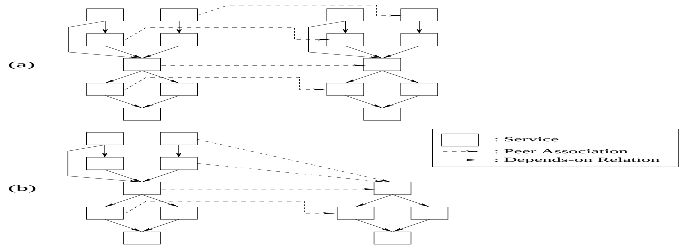

Figure 4: Two Communication Models: (a) The "peer to peer" and (b) the "firewall" model

of the services along the path. Thus, this is similar to P being an application entity that uses the services provided by the protocol reference model that comprises of the (sub-)path that beings at Q.

2.1 Communication Models

The models for communication we assume in this paper are of two types: the "peer to peer" model and the "firewall" model (see figure 4).

In the "peer to peer" model, there is a sender and a receiver each of which has an identical protocol reference model with the associated services at each layer. Peer entities within a layer at the sender and receiver communicate with each other using PDUs. This is the model of communication typically used between end–systems in a network.

The second model is appropriate when a *firewall* [14, 15, 57] is used. A firewall is a point of control[2] introduced in a network and comprises mechanisms to enforce a security policy on entering and leaving communication traffic [57]. The communication subsystem for a firewall only consists of a portion of the layers of a full protocol reference model. Yet, the highest layer in the firewall expects to make inferences about the PDUs transferred from higher layers in the protocol reference model at the sender. Figure 4 depicts the two scenarios.

3 Security Requirements, Services and Mechanisms

The aim of communication security is to protect data that represents or codes information during its transmission in computer networks and distributed systems [39]. With this in mind, in this section, we discuss typical security requirements from a protocol reference model in terms of the properties of the communicated data and the consumed network resources. We then list the security services considered for incorporation into protocols in this paper in section 3.2, this list being drawn from [27]. Though we do not show how or to what extent the services fulfill the requirements, we do give examples of the mapping between the services and requirements. We then discuss security mechanisms in section 3.3. An important goal of this section is to clarify the differences between security requirements, services and mechanisms. The rest of the paper concerns itself primarily with the security services. We also discuss existing work that addresses similar issues as we address in this paper in section 3.5.

3.1 Security Requirements

Traditionally, security requirements have centered around three objectives: confidentiality, integrity and availability [22]. These objectives are derived from *threat* and *trust models* for the environment in question [31]. A *threat* is any potential occurrence, malicious or otherwise, that can have an undesirable effect on the assets and resources [1]. The types of threats that correspond to the three objectives stated above are: the disclosure threat, the integrity threat and the denial of service threat [1]. *Trust* is

[2]The word "control" is used generally, and is not to be confused with its use in the context of communication services.

the level of confidence that a system will behave as expected [23]. Thus, a *threat model* is an abstract representation of the assets and resources in a system and the associated elements of threat as they pertain to those assets and resources. A *threat tree* [1, 59, 67] is an example of a threat model. A *trust model* associates each component of a system with a level of confidence of its proper functioning under various circumstances. Examples of trust models are the Bell-LaPadula disclosure model [1, 8], the Biba integrity model [1, 9] and the Clark–Wilson integrity model [1, 17].

In this paper, we express the typical security requirements from a protocol reference model using security properties on the data that is transmitted and the network resources that are consumed. We believe this will lead to a model for specification of security requirements that can be fulfilled by the security services. The data sent is D_S and the data received is D_R. An application entity injects D_S into the protocol reference model PS_S at the sender that is delivered to the application as D_R by the protocol reference model PS_R at the receiver. The security requirements can now be stated in terms of D_S, D_R and the network resources consumed in the transmission. The security requirements are that:

- The data satisfy the *integrity property*, that is, D_S does not undergo any unauthorized change in being delivered as D_R (typically, we require that $D_S = D_R$).
- The data satisfy the *confidentiality property*, that is, the information contained in D_S is only available to the sender and that in D_R is only available to the receiver.
- The network resources satisfy the *availability property*, that is, once D_S is sent by the sender, D_R should be made available to the receiver in "reasonable time."

It is important to emphasize that the requirements stated above are only "typical." They can be thought of as super–sets (or an overkill) of the security requirements that will be imposed on protocol reference models. There is a need for a more elaborate framework or model within which a user can specify her security requirements out of a protocol reference model.

Thus, the protocol reference model is required to fulfill the stated security requirements, that it does by enhancing the existing communication services with security services, which we discuss in the next section.

3.2 Security Services

A *security service* is a service (similar to a communication service) that attempts to fulfill a security requirement either partly or wholly. The OSI security architecture prescribes certain security services for incorporation into the OSI protocol reference model [22, 27, 39]. We will consider the same security services in this paper. These security services are:

- Authentication - An authentication service provides for confirming that a claimed identity is the true identity.
  - Peer Entity Authentication - A peer entity authentication service is to verify that a peer entity in an association is the one it claims to be [39]. In this context, an association is the same as a *connection*. A connection is an association between two peer entities having an establishment phase, a data transfer phase and a release phase [22]. We require that at least two of the three phases be non–trivial, that is, include the exchange of at least one PDU of non–zero length.
  - Data Origin Authentication - A data origin authentication service is to allow the sources of data received to be verified as claimed [39].
- Access Control - Access control services are to provide for the protection of system resources against unauthorized use [39].
- Confidentiality - A confidentiality service provides for secrecy of information from unauthorized parties in an information channel [20].

◦ Data Confidentiality - A data Confidentiality service makes it infeasible to deduce sensitive information from the content or size of a given data item [22]. In the ISO standard [27], data confidentiality is subdivided into the following:

▹ Connection Confidentiality - A connection confidentiality service is to provide confidentiality of all data transmitted in a connection [22, 39].

▹ Connectionless Confidentiality - A connectionless confidentiality service is to provide data confidentiality for all data comprising one connectionless data unit [22]. In this paper, one connectionless data unit is equivalent to one PDU.

▹ Selective Field Confidentiality - A selective field confidentiality service is to provide data confidentiality for specific field within the data during a connection or in a single data unit [39].

◦ Traffic Flow Confidentiality - A traffic flow confidentiality service makes it infeasible to deduce sensitive information by observing network traffic flows [22].

• Data Integrity - Data integrity services detect unauthorized change in data during transmission.

◦ Connection Integrity with Recovery - A connection integrity service with recovery is to provide for integrity of data in a connection. The loss of integrity is not only detected, but also recovered if possible [39].

◦ Connection Integrity without Recovery - A connection integrity service without recovery is to provide for the integrity of data in a connection [39]. Though the loss of integrity can be detected, it cannot be corrected.

◦ Selective Field Connection Integrity - A selective field connection integrity service is to provide integrity of specific fields within the data in a connection [39].

◦ Connectionless Integrity - A connectionless integrity service is to provide for the integrity of single data units [39], that is, a PDU.

◦ Selective Field Connectionless Integrity - A selective field connectionless integrity service is to provide for the integrity of specific fields within a single data unit [39], that is, a PDU.

The key difference between a connection integrity service and a connectionless service is that a connection integrity service has to able to detect data arriving out of sequence.

• Non–Repudiation - Non-repudiation services provide irrefutable evidence either that data was sent as claimed, was received as claimed or both [22].

◦ Non-repudiation with Proof of Origin - A non-repudiation service with proof of origin is to provide the recipient of a message with protection in a disagreement on whether a particular party originated a particular data item and/or disagreement about the time this origination occurred [22].

◦ Non-repudiation with Proof of Delivery - A non-repudiation service with proof of delivery is to provide the sender of a message with protection in a disagreement on whether a particular data item was delivered to a particular party and/or a disagreement about the time this delivery occurred [22].

The mapping of security services to security requirements is many to many. That is, a security requirement maps to several security services and a security service serves to fulfill more than one security requirement. For instance, the authentication services and the confidentiality services serve to fulfill the confidentiality requirement. This is because in the absence of an authentication service, it is possible that a third party receives data that was not intended for it, thus violating the confidentiality property (that is, D_S is the data that is sent and $D_R = \emptyset$, since the assumed receiver did not receive any data). Similarly, the data origin authentication service serves both towards the confidentiality requirement and the availability requirement.

Also, the list of security services and their corresponding definitions suggest that the services are not orthogonal to each other. Authentication can be thought of as a weaker form of non-repudiation. With authentication, we know (or have a confirmation of) who the other party is, with non-repudiation we are able to prove this to an impartial judge [31]. Similarly, access control is dependent on authentication, since controlling access to resources is based on the confirmed identities of entities attempting to use those resources [31].

3.3 Security Mechanisms

Security mechanisms are similar to the mechanisms we considered for "generic" communication services in section 2 and figure 3. Security mechanisms realize or implement one or more security services either partly or wholly. The ISO standard [27] enumerates eight specific security mechanisms [39]: Encipherment, Digital Signature Mechanisms, Access Control Mechanisms, Data Integrity Mechanisms, Authentication Exchange Mechanisms, Traffic Padding Mechanisms, Routing Control Mechanisms and Notarization Mechanisms. We refer the reader to [22, 39] for an exposition on the above mechanisms. The mapping between security services and security mechanisms is also many to many. For instance, encipherment using the Data Encryption Standard (DES) [10, 35] can be used for both the data origin authentication service and the data confidentiality services. Also, the data origin authentication service can be achieved either by encipherment using DES [10, 35] or digital signature mechanisms such as Digital Signature Algorithm (DSA) [10, 36], with message digest algorithms such as the Secure Hash Algorithm (SHA) [10, 37] and MD5 [10, 53] that are classified as data integrity mechanisms.

3.4 The Distinction between Requirements, Services and Mechanisms

The distinction between security requirements, services and mechanisms is crucial to the issues addressed in this paper. This distinction is not made clear in existing literature on this topic.

Security requirements, services and mechanisms can be thought of as three levels of abstraction, from highest to lowest. Security requirements are needs related to security. They are specified abstractly by a user. Security requirements involves identification of relevant entities in the communication environment, some of which are sources of threats. A threat model models the nature and extent of the threat perceived from each of the sources of the threats. A trust model models the extent of the trust an entity has in each of the other entities. A security requirement thus expresses the sufficient conditions in securing a communication subsystem in such an environment.

Security services are derived from security requirements and are functional components of a protocol reference model (like the communication services). They fulfill one or more of the security requirements. Similar to a communication service, a control, such as a finite state automaton, can be used to describe a security service at a lower level of abstraction. The controls that comprise a security service are realized in security mechanisms, which are protocols or algorithms.

In this paper, we choose to address issues at the level of abstraction of security services and the controls associated with them, that is, at a level of abstraction above specific mechanisms. In particular, in incorporating security mechanisms into an instance of a protocol reference model, it is conceivable that several variables will be added to the existing list of variables (see figure 3.) Such variables are beyond the scope of this paper.

3.5 Existing Work

There is a wealth of research work on the issue of incorporation of security services into protocol reference models. But, existing work typically focuses on a single model (such as OSI). Further, there is often no distinction made between security services and security mechanisms. Also, recall the distinction made in section 1 about two possible perspectives in the incorporation of security services into a protocol reference model. Existing literature takes the approach of fulfilling security requirements out of the relevant protocol reference model. Yet, the requirements are not formally specified, and therefore it is difficult to reason about the level of security provided.

[62, 63, 65, 66] are papers that attempt to formalize the specification of security requirements in the context of protocol reference models. But they do not provide a high level framework or model within which a user can specify security requirements. [62, 63] focus on the DoD Trusted Computer System Evaluation Criteria (TCSEC) [21] and adopt the perspective from [34] that a formal security policy model is required only for access control. Further, [62, 63] do not relate security requirements to security services in specific layers or protocols in a protocol reference model. [65, 66] attempt to apply the well known Biba integrity model [1, 9] and Bell–LaPadula [1, 8] disclosure model to the OSI environment. But the focus in both papers in on distributing security functionality throughout the OSI reference model while minimizing the number of keys that need to be used. In other words, there is no intermediate mapping from the requirements to the services, and thus, no requirements–based arguments are made for the incorporation of the services in the layers.

[31, 46, 47, 50, 51]) make recommendations for the incorporation of security services into the OSI protocol reference model [26]. As we mentioned before, these papers do not distinguish between services and mechanisms and deal mainly with the issue of handling keys (which are used in certain security mechanisms). While the choice of security mechanisms is also important, it must be distinguished from choosing security services such that the security requirements are satisfied. Also, these papers do not adopt a formal security requirements framework and therefore only informally argue for the validity of their recommendations. Similarly, [2, 3, 4, 48, 49] make recommendations for the placement of certain security services in the TCP/IP protocol suite and [16, 19, 42, 61, 64] make recommendations for such placement in the context of ATM. As we mentioned before, none adopts a formal requirements model and therefore only informally justify their choices. Further, there is no distinction between services and mechanisms.

4 Criterion for Placement

Our objective in this paper is not to argue about the appropriateness of the placement of security services as dictated by the security requirements. Instead, we adopt the second approach alluded to in section 1.2. That is, we merely want to make recommendations for which security services would be "compatible with" existing communication services.

So, in this section, we propose a criterion for the incorporation of the security services enumerated in section 3.2 into a protocol reference model as characterized in section 2. Our approach is that security services are to be integrated into a layer as characterized in section 2 and therefore are realized similar to the communication services. Thus, at a lower level of abstraction, a security service can be modeled using a Mealy machine. For each security service, we first identify the variables associated with it that don't tie the service with a specific mechanism. Thus, in our discussions that follow, for each of the security services considered, we first isolate those variables that are essential to our criterion for incorporation, but keep us from considering specific realizations of the service. Our criterion for incorporation is:

> *A security service can be incorporated into a layer if the set of variables associated with that (security) service is a subset of the union of the sets of variables of the (communication) services already provided at the layer.*

The reasoning behind the criterion is that security services are meant to enhance existing services in the layers. Thus, no redundancy is added by the incorporation of the security services.

A method for the incorporation follows directly from the criterion stated above. We isolate the variables needed to realize each security service (not the mechanisms) and check if the variables are already part of the services provided by the layer in question. If they are, then we okay this security service for incorporation into the layer, otherwise, we say that the security service is inappropriate for incorporation into the layer. Of course, once we state that a particular security service is appropriate for incorporation into a layer, there still remains the crucial issue of what assumptions need to be made about the rest of the protocol reference model for the provision of this security service, for the two models of communication adopted in section 2.1, especially keeping in mind the principles of layering that we listed in section 2.

In the upcoming sections of the paper, we isolate the appropriate variables for each of the security services discussed in section 3.2 and state the assumptions that need to be made in the rest of the protocol reference model in providing the service. Note that in adding the security services to a layer, additional variables will be added to the list of variables already associated with the layer. Thus, more than one iteration in applying the criterion may be needed to achieve the desired security.

4.1 Quality of the Variables

Since our criterion for incorporation of the security services from section 4 hinges on the variables associated with the existing services at a layer, the quality of a variable affects the quality of the security service that can be provided. For instance, non-repudiation (see section 3.2) of time of delivery requires the provision of an appropriate variable in the controls of the existing services pertaining to time (see section 9). Thus, the quality of the non–repudiation functionality is limited by the granularity of the time variable. Trust in the variable is another instance of the measure of quality in it. For non–repudiation, not only do we require a variable of fine enough granularity, but only one that is trustworthy. Of course, if the quality of the variable is enhanced, the quality of the corresponding security service is also enhanced.

5 Authentication

The two types of authentication we consider in this paper, that we mentioned in section 3.2 are peer entity authentication and data origin authentication.

Peer entity authentication pertains to ascertaining an identity in a connection. Therefore, any control that realizes a peer entity authentication service will involve the following an identity variable, and variables pertaining to a connection.

Thus, a peer entity authentication service only makes sense with a connection-management service [22, 60] where a variable of the services offered by the layer is an identity. For instance, multiplexing is a communication service that uses an identity as a variable since an identity would be needed to demultiplex data into the different destinations. Thus, to extend the example, a peer entity authentication service would be appropriate for a layer that offers the connection management and multiplexing services. An instance of such a layer is the transport layer in TCP/IP [43, 44]. The identity here is a port number.

Data origin authentication pertains to ascertaining the source of a PDU. Since we assume that data flows through the protocol stack as PDUs, the only variable that would make a data origin authentication service appropriate for a layer is an identity. Again, as an example, a layer that provides the multiplexing service would be appropriate for the incorporation of a data origin authentication service. Examples of such layers are the transport layer (port number), the internetworking layer (IP address) and the data link layer (data link address) in TCP/IP [43, 44]. The internetworking layer multiplexes traffic from several sources onto the internetwork, while the data link layer multiplexes traffic from several sources onto a local area network.

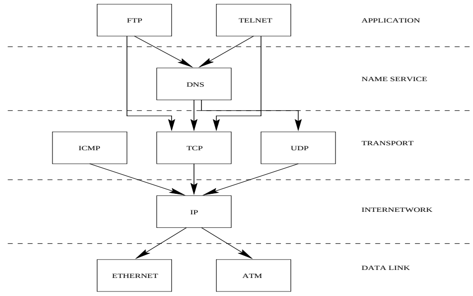

Figure 5: A Portion of the TCP/IP Protocol Suite

5.1 Assumptions in Providing an Authentication Service

In the communication involving two instances of a protocol reference model in which several of the layers offer a service that use an identity variable, a communication instance can be characterized by the combined identities maintained in those layers. As an example, figure 5 depicts the use of the File Transfer Protocol (FTP) [45] over the TCP/IP protocol suite with Ethernet at the data link layer. A particular instance of such a communication can be identified by the combined identities at each layer. In the example from figure 5, the combined identity (*port number*, *internetworking address*, *data link address*) identifies a particular instance of a communication.

In keeping with the principles of layering we listed in section 2, a layer such as the transport layer is the "representative" for the rest of the protocol reference model (that is, the layers below) to the application layer. Since the mapping from an SDU to a PDU (and vice versa) in an entity is service and mechanism dependent, a protocol such as TCP is unable to assume anything about the mapping in layers below it. In other words, TCP is unaware of how protocols at layers beneath it generate PDUs or manage connections. Thus, in providing an authentication service to a layer above it, the transport layer is forced to assume that in all layers below it in which authentication is appropriate (that is, layers in which a service has an identity variable), authentication is done.

Thus, the assumption that needs to be made at a layer that provides the (peer entity or data origin) authentication service is that such authentication is also performed in all layers below it for which the authentication service is appropriate. The protocol cannot enforce such behaviour since the controls and variables in the other protocols are invisible to this protocol. Note that this assumption depends on the security requirements. If the security requirements reflect that authentication at some of the lower layers is not necessary, then that would be the overriding factor that determines provision of the authentication service.

The provision of authentication for the "firewall" model for communication is discussed in [56], which calls for a potential violation of the layering principles we adopted in section 2 by providing for an *entity association* between entities that are not in adjacent layers and *information sharing* between them for efficiency. Thus, for the "firewall" model for communication, we need to make additional assumptions about cooperation between entities that are not necessarily in adjacent layers. In keeping with the principles of layering adopted in section 2, we need to assume that the intermediate layers provide support (through appropriate services) for such entity association.

6 Access Control

As we remarked in section 3.2, access control is closely tied to authentication. Thus, the variable that a control that realizes the access control service would be based on is an identity, as happens with authentication. Thus, the access control service is appropriate for incorporation into a layer that has at least one existing service that employs an identity variable. Examples of such layers are the transport and the data link layers. An additional point to note here is that access control based on a certain identity variable has to be performed at or above the layer in which that identity is used. For instance, access control based on port number in figure 5 must be done at or above the transport layer. But, since the identity variable is only visible to the transport layer, it is appropriate to do the access control service within the transport layer.

6.1 Assumptions in Providing an Access Control Service

Access control is different from authentication in that the authentication services are primarily intended to fulfill the security requirements stemming from the confidentiality and integrity properties on the data that is transmitted. Access control is primarily intended to fulfill the security requirement(s) arising out of the availability property of the network resources. Thus, again depending on the security requirements, a lower layer in a reference model has to assume that the layers above it do access control (if so deemed by the security requirements.) This is the "reverse" of the assumption that needs to be made for authentication. Thus, for the "firewall" model for communication, we do not have to make any additional assumptions (in terms of entity association between entities belonging to non–adjacent layers, for instance.)

7 Confidentiality

As we remarked in section 3.2, the aim of a confidentiality service is to keep information secret. In the context of this paper, if the information pertains to the content (both syntax and semantics) or size of the data, we consider that data confidentiality, and if the information pertains to traffic characteristics (such as data rate), we consider that traffic flow confidentiality.

We assumed in section 2 that data would flow through the instance of a protocol reference model as PDUs. That is, through every entity engaged in the communication, PDUs flow over time. Thus, for a connectionless confidentiality service, the set of variables that determine the appropriateness of incorporation of the confidentiality service is empty and the service would be appropriate for any protocol. For the connection confidentiality service, the set of variables needed to realize it would be the set of variables associated with a connection. Thus, it is appropriate for any layer that performs the connection management service. As for the selective field confidentiality service, it is necessary for the service at a certain layer to be aware of the format of the portion of the PDU to which the service is to be applied. This would be the case if that portion of the PDU is generated by the entity. Thus, a selective field confidentiality service is appropriate to any layer, but only if the field the confidentiality service is used for is generated by that layer.

Traffic flow confidentiality can also be provided at any layer, since PDUs flow through each entities engaged in the communication. In other words, the set of variables needed to realize traffic flow confidentiality is the empty set. Note that provision of the traffic flow confidentiality service at a layer implies that the traffic parameter(s) are made secret from all the lower layers in the communication.

7.1 Assumptions in Providing Confidentiality

The assumption that needs to be made in providing data confidentiality is that the syntax of the portion of the PDU over which the service is being provided must be known at the layer in question. Since we do not specify that the syntax of a PDU is invisible to layers other than that of the entity in question as part of our layering principles, it is conceivable that a layer below the one that generates the data provides for the data confidentiality service of that data. As an example, in the OSI protocol reference model [26], the presentation layer performs a translation service for the SDUs received from the application layer. Thus, if the data confidentiality service is a service provided at the application layer, and it obfuscates the syntax or format of the SDUs at the presentation layer, the translation service cannot be provided. But, if we assume that the syntax of the PDUs handed down by the entity at the application layer is known to the presentation layer, the presentation layer can still adapt its translation as required. Even better would be to provide the data confidentiality service in the presentation layer and not the application layer.

The assumption to be made with respect to traffic flow confidentiality is more subtle. Since a layer that provides such a service seeks to keep traffic parameters of the upper layers secret from all entities below it on the protocol reference model (and the network), such a layer must assume that those layers above it do not require access to such parameters. For example, in figure 5, the transport layer provides flow and congestion control services. If a layer below the transport layer, such as the internetworking or the data link layer, provided a traffic flow confidentiality service, this might impinge on the effect the transport layer's flow and congestion control services attempt to have on the traffic parameters. Thus, an assumption to be made in providing the traffic flow confidentiality service within a layer is that layers above it don't need access to the values of the same parameters from below this layer that the confidentiality service is being provided for.

8 Data Integrity

Provision of data integrity is similar to the provision of data confidentiality in that the syntax of the PDU may need to be known to provide the service. This would imply that the service is most appropriate for the layer in which the portion of the PDU over which the service is applied to is generated. But, similar to the case with the data confidentiality services, we do not assume that the syntax or format of a PDU is only visible within the layer it is generated in. Also, for connection integrity, a control realizing it has to maintain state across PDUs, since the order in which the PDUs are delivered needs to be maintained.

Thus, the set of variables that need to be present in the existing services in a layer for connection integrity with recovery, connection integrity without recovery and selective field connection integrity are the variables associated with a connection and a variable associated with the sequence of the PDUs, such as a sequence number. Thus, they are appropriate for incorporation layers that provide the connection management service, that maintain one or more variables that retain sequencing information. The set of variables for connectionless integrity and selective field connectionless integrity is the empty set and there are appropriate for incorporation into any layer.

8.1 Assumptions in Providing Data Integrity

As with data confidentiality the assumption that needs to be made in providing for integrity is that the syntax or the format of the (portion of the) PDU(s) over which the service is applied needs to be known at the layer in question for a data integrity service to be provided.

9 Non–Repudiation

As we remarked in section 3.2, non-repudiation is closely tied to authentication. A control that realizes the non-repudiation service is centered around the identity of the sender, in the case of non-repudiation with proof of origin, and the receiver, for non-repudiation with proof of delivery. Since the sender needs assurance that the data was received as claimed or the receiver that the data received was indeed what was sent, non-repudiation also overlaps with the data integrity services. Further, since the time of sending and/or delivery is important, there is also an overlap with the access control service.

Thus, the variables that must be part of the existing services in a layer for the provision of non-repudiation with proof of origin is the identity of the sender and for non-repudiation with proof of delivery is the identity of the receiver. Also, the other variables involved in the non–repudiation, such as a time–stamp, must be part of the existing services. We discuss the dependency of non-repudiation on integrity and access control as an assumption in the next section.

9.1 Assumptions in Providing Non–Repudiation

The assumptions in providing non-repudiation are that data integrity is also provided within the layer that non-repudiation is provided in and that access control is enforced on the network resources on other senders of data so that the time criterion of non-repudiation can be met. Further, as with authentication, depending on the environment of operation as gleaned from the security requirements, a layer that offers the non-repudiation service may need to assume that non-repudiation services are offered in all layers below it in the model that have an identity variable as a variable associated with the existing services.

10 Examples

In this section, we use existing protocol reference models from TCP/IP and ATM and apply our criterion for the placement of the security services from section 3.2. We also incorporate the assumptions stated for each service in previous sections in our placement. Further, we compare our results to recommendations from the literature. As we remarked in section 1, existing work typically recommends where security services should be placed. Thus, we only check if the recommendation in question is a possibility suggested by our method.

10.1 TCP/IP

We have already used a portion of the protocol reference model from TCP/IP as an example in clarifying some of our comments in the sections that pertain to the placement of the security services in a protocol reference model in general. We present those and other results in this section. We also compare our results to the IP Security Protocols (IPSEC) [2, 3, 4]. The portion of the protocol reference model we focus on is the one in figure 5.

10.1.1 Authentication

Our analysis concludes that it is appropriate to incorporate peer entity authentication into the transport layer and perhaps the data link layer since they provide connection management and have identity variables associated with them. Data origin Authentication is appropriate to the application, transport, internetworking and data link layers, since they involve an identity each.

In keeping with our assumptions, provision of an authentication service at the transport layer implies that an authentication service is also being provided at the internetworking layer, and provision of an authentication service at the internetworking layer implies that an authentication service is provided at the data link layer.

10.1.2 Access Control

Access control also involves an identity variable and therefore it is appropriate for incorporation into all the layers for which data origin authentication is appropriate.

10.1.3 Confidentiality

Connection data confidentiality is appropriate to the transport and data link layers if they provide the connection management service. Connectionless confidentiality is appropriate to all the layers. We recall that there are no prerequisite variables in the incorporation of connectionless data confidentiality. As for selective field confidentiality, since TCP/IP is built with the philosophy that the format of the SDU is not known at a layer the SDU is provided as input to, it should be performed at the layer that the relevant portion of the PDU is constructed in. Thus, selective field confidentiality is appropriate to all the layers, but only for the data generated at those layers.

Traffic flow confidentiality is not appropriate for the internetwork or data link since the transport layer involves flow and congestion control services. The provision of flow and congestion control at the transport layer could be adversely affected by traffic flow confidentiality services if they are offered at those layers. It would be appropriate to provide for traffic flow confidentiality at the transport layer or a layer above it. The advantage with providing the service at the transport layer is that traffic parameters on the PDUs generated at the transport layer can also be incorporated into the service.

10.1.4 Data Integrity

The connection integrity with recovery, connection integrity without recovery and selective field connection integrity services are appropriate to the transport and data link layers given that they provide a connection management service. The connectionless integrity and selective field connectionless integrity services are appropriate to all the layers.

10.1.5 Non-repudiation

The non-repudiation services are appropriate to all the layers except the name–service layer, if data integrity and access control mechanisms are in place for those layers. Further, in keeping with our assumptions, provision of a non–repudiation service at the internetwork layer implies that it is also provided at the data link layer and provision of a non–repudiation service at the transport layer implies that it is also provided at the internetwork layer.

10.1.6 Comparison to IPSEC

We expect our results to be a super–set of any recommendation that attempts to fulfill security requirements. In other words, if our recommendations do not deem a security service to be appropriate to a layer, then we expect that a recommendation targeting fulfillment of security requirements will also not deem such a service to be appropriate to the protocol or layer in question.

The IP Security Protocols (IPSEC) [2, 3, 4] attempt to partly fulfill informally stated security requirements in the internet. They call for the incorporation of Encapsulating Security Payload (ESP) and Authentication Header (AH) security mechanisms as part of IP. These are intended to implement the provision of data origin authentication and connectionless data confidentiality security services at the internetwork layer. Thus, only a small subset of the services considered in this paper are considered in IPSEC.

Based on what we discussed above, incorporation of data origin authentication and connectionless data confidentiality at the internetwork layer is consistent with our results.

10.2 ATM

In this section, we apply our analysis on the incorporation of security services into the layers of the ATM Protocol Reference Model [40] and compare our results to the suggestions made in the ATM Forum's security specification [61].

ATM is a technology or suite of protocols that offers connection–oriented, switched communication services based on virtual circuits. A key feature of ATM is that the network provides guaranteed Quality of Service (QoS) to the user. The initial design of the protocol reference model for ATM did not incorporate security requirements. Therefore, there is now a pressing need to enhance the protocol reference model for ATM by incorporating security services into it.

10.2.1 The ATM Protocol Reference Model and Protocols Used

The protocol reference model for ATM is shown in figure 6 [12, 40]. The reason the protocol reference model is displayed as a three dimensional box is that the functionality associated with the *user*, *control*

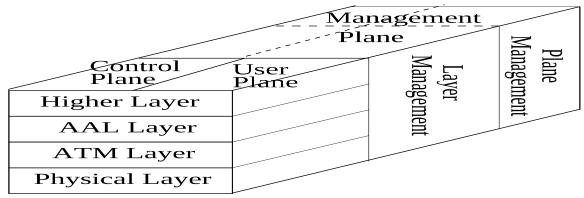

Figure 6: The ATM Protocol Reference Model

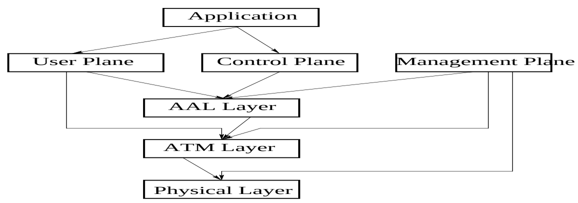

Figure 7: The Two-Dimensional Version of the ATM Protocol Reference Model

and *management planes* is orthogonal to that of the *AAL*, *ATM* and *physical* layers. We adapt the ATM protocol reference model to the general model suggested by figure 2 by viewing it as in figure 7.

The functions performed by each of the layers is indicated in figure 8. The functionality of each of the layers (and planes) is realized with the various protocols that are used at the layers. The protocol used for connection management at the control plane is the User Network Interface (UNI) protocol [5]. The protocol used for routing at the control plane is the Private Network–Node Interface (PNNI) protocol[3]. The protocol used at the management plane is the point to point Integrated Local Management Interface (ILMI) protocol [7]. Various protocols are used at the AAL layer: AAL1, AAL2, AAL3/4 and AAL5. The signaling protocol, UNI, also uses the Service Specific Connection Oriented Protocol (SSCOP) [13] for reliable transmission. SSCOP is similar to TCP, but is only intended for point–to–point reliable transmission services.

In the next section, we begin to consider placement of the security services we discussed in section 3.2 in the ATM protocol reference model. As we apply our criterion for placement, we also introduce additional details of the layers and their associated services.

10.2.2 Authentication, Access Control and Non–Repudiation

As we discussed in section 3.2, authentication, access control and non–repudiation are related services. Therefore, we discuss their placement jointly. We refer to figure 7 in our considerations for placement, but exclude the application layer from consideration.

As we indicated in sections 5, 6 and 9, an identification variable is a prerequisite in any layer for the incorporation of authentication, access control and non–repudiation services. The control plane deals with (end–to–end) ATM addresses during connection setup and therefore it is appropriate to incorporate data origin authentication into the control plane. The protocol that realizes the routing service organizes participating entities[4] into "peer groups," and entities in a peer group require point–to–point identification. Thus, data origin authentication is also appropriate to the control plane. Similarly, the ATM layer deals with virtual circuit identifiers and therefore data origin authentication is appropriate at the ATM layer as well. But data origin authentication at the ATM layer would only have point–to—point significance. The management plane's service uses an identification variable to identify itself to

[3] We are only considering private ATM networks in this paper. The corresponding protocols used between private and public ATM networks and within public networks are the public UNI and the Broadband Inter–Carrier Interchange (B–ICI) [6] protocols.

[4] Participating entities in PNNI are private ATM switches.

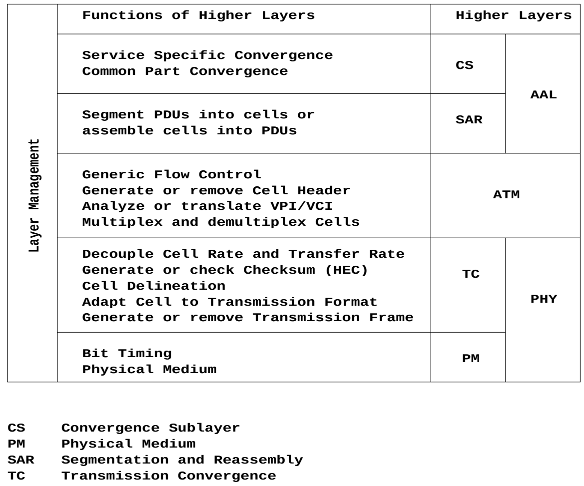

Figure 8: Functions Performed at the Layers in the ATM Protocol Reference Model

a peer. Thus, data origin authentication is appropriate to the management plane also. Data origin authentication is not appropriate to any of the other layers.

Peer entity authentication is only appropriate to those layers that involve connection management. ATM is connection–oriented although some connections only have point–to–point significance. Thus, peer entity authentication is appropriate to all layers that data origin authentication is appropriate to and thus, peer entity authentication is appropriate to the control plane, management plane and the ATM layer.

Access control and non–repudiation are based on similar principles as authentication and therefore those services are appropriate to all layers that authentication is appropriate to.

Note that the assumption in providing authentication is that such a service also be provided at all layers below that are appropriate for such a service. This can of course be overridden by the security requirements. But, in general, for ATM, the control plane and management planes will assume that (point–to–point) data origin and peer entity authentication is provided at the ATM layer. Similarly, for access control to be appropriate to the ATM layer, such access control must also be provided at the control and management planes.

10.2.3 Confidentiality

Following our remarks in section 7, data confidentiality can be provided at any (or all) of the layers in the ATM protocol reference model. Selective field confidentiality can only be provided if the syntax of the PDUs are known at the layer in question. Thus, though the service can be provided anywhere in the protocol reference model, it should only be applied to the data generated at the layer in question. Traffic flow confidentiality assumes that a communication (or security) service at a higher layer does not need access to traffic parameters. Thus, since the ATM layer provides a traffic policing functionality [40], traffic flow confidentiality should only be provided at or above the ATM layer.

10.2.4 Integrity

Since ATM is connection–oriented, there is no need to distinguish between connection integrity and connectionless integrity. If a service is appropriate to a layer, so is the other. Integrity services can be provided at any layer in the protocol reference model. But, similar to the case with confidentiality, the syntax of the portion of the PDUs the service is applied to must be known at the time of use of the service.

10.2.5 Comparison to the Recommendations of the ATM Forum

The ATM Forum's security specification [61] gives an informal specification of the security requirements in an ATM network. It then makes recommendations for placement of the security services. Not all security services we consider in this paper are considered in the specification. Specifically, peer entity authentication, data origin authentication, data confidentiality and data integrity are the services considered in the specification. The focus is on security mechanisms and no arguments are given about the appropriateness of a service at a certain layer in the protocol reference model.

[61] recommends that data confidentiality be provided at the ATM layer. This is consistent with our recommendation. Of course, [61] also bases its recommendations on criteria not considered in this paper (such as efficiency). [61] recommends that peer entity authentication be performed either at the user plane or at the control plane. Though performing peer entity authentication at the control plane is consistent with our recommendation, providing that service at the user plane is not. The specification calls for the user plane to adopt an end–to–end perspective in providing such a service. We believe that this could lead to violations of the ATM protocol reference model. The specification also calls for data origin authentication to be performed at the AAL layer with the data integrity service. Though providing the data integrity service at the AAL layer is consistent with out recommendations, providing data origin authentication is not. The reason is that the AAL layer provides services that do not involve any identification. We believe that provision of the data confidentiality service at the AAL layer could also lead to violations of the ATM protocol reference model.

11 Conclusions

This paper provides new insights into the issue of incorporating security services into the layers of a protocol reference model by considering the problem at a level of abstraction not explored in existing work. The discussions culminate in a concrete criterion to base the incorporation of security services into a layer on. The criterion hinges on examining the existing communication services at a level of abstraction of controls and their associated variables, and isolating the corresponding variables in the security service that needs to be incorporated. The paper also presents two examples of the application of the criterion, and compares and contrasts the results with existing recommendations.

Thus, the main contributions of this paper are:

- Provides new insights into the issue of incorporating security services into the layers of a protocol reference model by considering the problem at a level of abstraction not addressed in existing work.
- Describes a criterion to base the incorporation of a security service into a layer on.
- Validates the criterion by applying it to two examples and comparing and contrasting the results to existing recommendations.

Acknowledgements

We would like to thank Christoph Schuba at Sun Labs for a review of an earlier draft of this paper and for making several insightful comments. Several changes based on those comments are reflected in this draft.

References

[1] Edward Amoroso. *Fundamentals of Computer Security Technology.* Prentice Hall, 1994.

[2] R. Atkinson. *RFC-1825 Security Architecture for the Internet Protocol.* Information Sciences Institute, USC, CA, August 1995.

[3] R. Atkinson. *RFC-1826 IP Authentication Header (AH).* Information Sciences Institute, USC, CA, August 1995.

[4] R. Atkinson. *RFC-1827 IP Encapsulating Security Payload (ESP).* Information Sciences Institute, USC, CA, August 1995.

[5] ATM Forum. *ATM User-Network Interface Specification, Version 3.1.* Prentice-Hall, Englewood Cliffs, New Jersey, September 1994.

[6] ATM Forum. *ATM Broadband Inter–Carrier Interchange Specification Version 2.0.* ATM Forum Technical Committee, December 1995.

[7] ATM Forum. *ATM Integrated Local Management Interface (ILMI) Specification Version 4.0.* ATM Forum Technical Committee, September 1996.

[8] D. Bell and L. LaPadula. *Secure Computer Systems: Mathematical Foundations, ESD-TR-73-278, Vol. 1.* Mitre Corporation, 1973.

[9] K. Biba. *Integrity Considerations for Secure Computer Systems, MTR-3153.* Mitre Corporation, 1975.

[10] Bruce Schneier. *Applied Cryptography.* John Wiley and Sons, Inc., second edition, 1996.

[11] S. Budkowski and P. Dembinski. An Introduction to Estelle: A Specification Language for Distributed Systems. *Computer Networks and ISDN Systems*, 14, 1987.

[12] CCITT. *I.321 B-ISDN Protocol Reference Model and its Application.* CCITT, 1991.

[13] CCITT. *Q.2110 B-ISDN SAAL Service Specific Connection Oriented Protocol (SSCOP).* CCITT, 1994.

[14] D. Brent Chapman and Elizabeth D. Zwicky. *Building Internet Firewalls.* O'Reilly and Associates, Inc., 1995.

[15] William R. Cheswick and Steven M. Bellovin. *Firewalls and Internet Security - Repelling the Wily Hacker.* Addison-Wesley Professional Computing Series, 1994.

[16] Shaw-Cheng Chuang. Securing ATM networks. *Journal of Computer Security*, 4, 1996.

[17] D. Clark and D. Wilson. A Comparison of Commercial and Military Computer Security Policies. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1987.

[18] D. E. Comer. *Internetworking with TCP/IP.* Prentice-Hall, third edition, 1995.

[19] Robert H. Deng, Li Gong, and Aurel A. Lazar. Securing data transfer in asynchronous transfer mode networks. In *Proceedings Globecom '95*, November 1995.

[20] D.E. Denning. *Cryptography and Data Security.* Addison Wesley, Reading, MA, 1982.

[21] Department of Defence. *DoD Trusted Computer System Evaluation Criteria: DoD 5200.28–STD.* DoD, December 1985.

[22] Warwick Ford. *Computer Communications Security: Principles, Standard Protocols and Techniques.* Prentice Hall, 1994.

[23] Simson Garfinkel and Gene Spafford. *Practical UNIX and Internet Security.* O'Reilly and Associates, Inc., second edition, April 1996.

[24] Fred Halsall. *Data Communications, Computer Networks and Open Systems.* Addison–Wesley Publishing Company, Inc., fourth edition, 1996.

[25] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley Publishing Company, Inc., 1979.

[26] ISO/IEC 7498. *Information Technology - Open Systems Interconnection - Basic Reference Model.* American National Standards Association, New York, 1984.

[27] ISO/IEC 7498-2. *Information Technology - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture.* American National Standards Association, New York, 1988.

[28] ISO/TC97/SC21/WG1/DIS9074. *Estelle - A Formal Description Technique Based on an Extended State Transtition Model.* American National Standards Association, New York, 1987.

[29] Bijendra N. Jain and Ashok K. Agrawala. *Open Systems Interconnection: Its Architecture and Protocols.* McGraw–Hill Series on Computer Communications, 1993.

[30] Arto Karila. VOPS - A Portable Protocol Development and Implementation Environment. In *Proceedings of the IFIP TC 6 Conference on Information Network and Data Communication*, pages 19–34, 1987.

[31] Arto T. Karila. *Open Systems Security - an Architectural Framework.* Telecom Finland, 1991.

[32] Dennis MacKinnon, William McCrum, and Donald Sheppard. *An Introduction to Open Systems Interconnection.* Computer Science Press, 1990.

[33] H. Mehrpour and A. E. Karbowiak. Modeling and Analysis of DOD TCP/IP Protocols using Numerical Petri Nets. In *Proc. 1990 IEEE Region 10 Conference on Computer and Communication System, TENCON'90*, pages 617–622, September 1990.

[34] National Computer Security Centre. *Trusted Network Interpretation of the TCSEC: NCSC–TG–005.* NCSC, July 1987.

[35] National Institute of Standards and Technology. *NIST FIPS PUB 46-2, Data Encryption Standard (DES).* U.S. Department of Commerce, December 1993.

[36] National Institute of Standards and Technology. *NIST FIPS PUB 186, Digital Signature Standard.* U.S. Department of Commerce, May 1994.

[37] National Institute of Standards and Technology. *NIST FIPS PUB 180-1, Secure Hash Standard.* U.S. Department of Commerce, April 1995.

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

[39] Rolf Opplinger. *Authentication Systems for Secure Networks.* Artech House, Inc., 1996.

[40] Othmar Kyas. *ATM Networks.* International Thomson Computer Press, second edition, 1996.

[41] Larry L. Peterson and Bruce S. Davie. *Computer Networks - A Systems Approach.* Morgan Kaufmann Publishers, Inc., 1996.

[42] Mohammed Peyravian and Thomas D. Tarman. Asynchronous Transfer Mode security. *IEEE Network*, May 1997.

[43] J. Postel. *RFC-791 Internet Protocol.* Information Sciences Institute, USC, CA, September 1981.

[44] J. Postel, editor. *RFC-793 Transmission Datagram Protocol.* Information Sciences Institute, USC, CA, September 1981.

[45] J. Postel and J. Reynolds. *RFC-959 File Transfer Protocol (FTP).* Information Sciences Institute, USC, CA, October 1985.

[46] Raju Ramaswamy. Application of a Key Generation and Distribution Algorithm for Secure Communication in Open Systems Interconnection Architecture. In *Proc. 1989 International Carnahan Conference on Security Technology*, pages 175–180, October 1989.

[47] Raju Ramaswamy. Placement of Data Integrity Security Services in Open Systems Interconnection Architecture. *Computers and Security*, 8, 1989.

[48] Raju Ramaswamy. Security Architecture for Data Transfer through TCP/IP Protocols. *Computers and Security - International Journal*, 8(8):709–720, December 1989.

[49] Raju Ramaswamy. A Security Architecture and Mechanism for Data Confidentiality in TCP/IP Protocols. In *Proc. 1990 IEEE Symposium on Research in Security and Privacy*, pages 249–259, May 1990.

[50] Raju Ramaswamy. Nonrepudiation Security Issues in OSI Computer Networks. In *Proc. 1990 IEEE NORTHCON'90 Conference*, October 1990.

[51] Raju Ramaswamy. Traffic Flow Confidentiality Security Service in OSI Computer Network Architecture. In *Proc. 1990 IEEE Region 10 Conference on Computer and Communication System, TENCON'90*, pages 24–27, September 1990.

[52] R. Rivest. *RFC-1320 The MD4 Message-Digest Algorithm.* Network Working Group, April 1992.

[53] R. Rivest. *RFC-1321 The MD5 Message-Digest Algorithm.* Network Working Group, April 1992.

[54] Harry Rudin. An Informal Overview of Formal Protocol Specification. In *Proceedings of the IFIP TC 6 Conference on Information Network and Data Communication*, pages 5–18, 1987.

[55] R. Saracco and P. A. J. Tilanus. CCITT SDL: Overview of the Language and its Applications. *Computer Networks and ISDN Systems*, 13:65–74, 1987.

[56] Christoph L. Schuba. *Distributed Functions in Heterogeneous, Layered Systems.* Ph.D Research Proposal, COAST Lab, Dept. of Computer Sciences, Purdue University, September 1995.

[57] Christoph L. Schuba. *On the Modeling, Design and Implementation of Firewall Technology.* PhD Thesis, Department of Computer Sciences, Purdue University, December 1997.

[58] K. S. Sivanandan, K. Garg, and N. K. Nanda. On the Petrinet Modeling of Medium Access Protocols. In *Proc. 1990 IEEE Region 10 Conference on Computer and Communication System, TENCON'90*, pages 628–632, September 1990.

[59] S. Smith. LAVA's Dynamic Tree Analysis. In *Proceedings of the 12th National Computer Security Conference*, 1989.

[60] William Stallings. *Data and Computer Communications.* Macmillan Publishing Company, fourth edition, 1994.

[61] Thomas D. Tarman, editor. *ATM Security Specification Version 1.0 (Draft).* ATM Forum Technical Committee, September 1997.

[62] V. Varadharajan. A Multilevel Security Policy for Networks. *IEEE INFOCOM '90*, June 1990.

[63] V. Varadharajan. Network Security Policy Models. In *Advances in Cryptology - AUSCRYPT '90*, pages 74–95. IEEE Comput. Soc. Press, 1990.

[64] Vijay Varadharajan, Rajan Shankaran, and Michael Hitchens. Security Issues in Asynchronous Transfer Mode. In *Proc. second Australasian Conference on Information Security and Privacy*, 1997.

[65] J. Verschuren, R. Govaerts, and R. Vandewalle. Realization of the Bell-LaPadula Security Policy in an OSI Distributed System Using Asymmetric and Symmetric Cryptographic Algorithms. In *Proceedings of the Computer Security Foundations Workshop V*, pages 168–78. IEEE Comput. Soc. Press, 1992.

[66] J. Verschuren, R. Govaerts, and R. Vandewalle. Simultaneous Enforcement of the Bell-LaPadula and the Biba Security Policy Models in an OSI Distributed System. *Communications on the Move, ICCS/ISITA 1992, C.S. Ng, T.S. Yeo and S.P. Yeo (eds)*, 1, 1992.

[67] J. Weiss. A System Security Engineering Process. In *Proceedings of the 14th National Computer Security Conference*, 1991.