

NCS TIB 05-2



NATIONAL COMMUNICATIONS SYSTEM

TECHNICAL INFORMATION BULLETIN 05-2

**IEEE STANDARD 802.1X for LOCAL and
METROPOLITAN AREA NETWORKS
PORT-BASED NETWORK ACCESS
CONTROL**

March 2005

**NATIONAL COMMUNICATIONS SYSTEM
Technology and Programs Division (N2)
PO Box 4502
Arlington, Virginia 22204-4502**

IEEE STANDARD 802.1X for LOCAL and
METROPOLITAN AREA NETWORKS
PORT-BASED NETWORK ACCESS CONTROL



Office of the Manager
National Communications System

March 2005

By
Communication Technologies, Inc.
14151 Newbrook Drive, Suite 400
Chantilly, Virginia 20151
703-961-9088 (Voice)
703-961-1330 (Fax)
www.comtechnologies.com

NCS TECHNICAL INFORMATION BULLETIN 05-2

**IEEE STANDARD for LOCAL and METROPOLITAN NETWORKS
PORT-BASED NETWORK CONTROL**

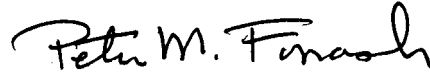
March 2005

PROJECT OFFICER:



DALE BARR, JR.
Chief, Advanced Technology
Technology and Programs Divisions

APPROVED FOR PUBLICATION:



PETER M. FONASH, Ph.D.
Deputy Manager
National Communications System

FOREWORD

Among the responsibilities assigned to the Office of the Manager, National Communications System (NCS) is the management of the Federal Telecommunications Standards Program. Under this program, the NCS, with the assistance of the Federal Telecommunications Standards Committee, identifies, develops, and coordinates proposed Federal Standards which contribute either to the interoperability of functionally similar Federal telecommunications systems or to the achievement of a compatible and efficient interface between computer and telecommunications systems. In developing and coordinating these standards, a considerable amount of effort is expended in initiating and pursuing joint standards development endeavors with appropriate technical committees of the International Organization for Standardization, the International Telecommunication Union-Telecommunications Standardization Sector, and the American National Standards Institute. This Technical Information Bulletin presents an overview of an effort that contributes to the development of compatible Federal and national standards in the area of network access control and is meant to inform interested Federal and industry parties. Any comments, inputs, or statements of requirements which could assist in the advancement of this work are welcome and should be addressed to:

National Communications System
Technology and Programs Division (N2)
PO Box 4052
Arlington, Virginia 22204-4052

**IEEE STANDARD for LOCAL and METROPOLITAN AREA NETWORKS
PORT-BASED NETWORK ACCESS CONTROL**

Abstract

This Technical Information Bulletin (TIB) examines the Institute of Electrical and Electronics Engineers (IEEE) 802.1X Standard for Local and Metropolitan Area Networks for Port-Based Network Access Control and its development. It discusses the definition of Extensible Authentication Protocol (EAP) and how it relates to both wired and wireless networks. It also compares the three most commonly used non-proprietary EAPs: Transport Layer Security (TLS) protocol, Tunneled Transport Layer Security (TTLS) protocol, and Protected Extensible Authentication Protocol (PEAP). Finally, the TIB presents conclusions and recommendations reflecting state of the art in EAP development and provides recommendations for continuing study and additional potential research.

Table of Contents

1	Introduction.....	1
1.1	Ethernet and Token Ring/FDDI Networks	1
2	Description of IEEE 802.1X	3
2.1	Scope.....	3
2.2	Purpose.....	4
2.3	802.1X Principles of Operation	4
2.3.1	Systems, Ports, and System Roles	5
2.3.2	Port Access Entity.....	5
2.3.3	Controlled and Uncontrolled Access	5
2.4	EAP Encapsulation over LANs (EAPOL).....	10
2.4.1	Transmission and Representation of Octets.....	11
2.4.2	EAPOL Frame Format for 802.3/Ethernet.....	11
2.4.3	EAPOL Frame Format for Token Ring/FDDI.....	11
2.4.4	Key Descriptor.....	12
3	Extensible Authentication Protocol (EAP).....	15
3.1	Point to Point Protocol.....	15
3.1.1	PPP Elements.....	15
3.1.2	Physical Layer Requirements	16
3.1.3	PPP Link-Control Protocol	17
3.2	Point-to-Point Protocol Extensible Authorization Protocol.....	18
3.2.1	Authorization Exchange.....	18
3.2.2	Request and Response Messages	19
3.3	Deploying EAP in a Corporate/Government Environment	21
4	Commonly Used EAP Methods.....	23
4.1	Transport Layer Security (TLS).....	23
4.1.1	TLS Functionality	24
4.1.2	TLS Operation	24
4.2	Tunneled Transport Layer Security	25
4.2.1	Phase 1: The TTLS Handshake.....	26

4.2.2 Phase 2: The TTLS Tunnel	26
4.3 Protected Extensible Authentication Protocol	27
4.3.1 Phase 1: The PEAP Handshake	29
4.3.2 Phase 2: The PEAP Tunnel.....	29
4.4 Comparison of TLS, TTLS, and PEAP	30
5 Security Applications of IEEE 802.1X.....	33
5.1 Ethernet and Token Ring/FDDI Networks	33
5.2 802.11 Wireless Networks.....	33
5.3 Virtual Private Networks	34
6 Observations and Summary	37
7 Recommendations.....	39
Appendix A: Acronyms	41
Appendix B: References	45
Appendix C: Bibliography.....	47

List of Figures

Figure 1-1 Typical Ethernet Network Topology	2
Figure 1-2 Typical Token Ring Network [14].....	2
Figure 1-3 Typical FDDI Network [15].....	2
Figure 2-1 802 Family of Standards for Local Area Networks [1].....	3
Figure 2-2 Uncontrolled and Controlled Ports [1].....	6
Figure 2-3 Effect of Authorization State on Controlled Ports [1]	7
Figure 2-4 Effect of MAC Enable/Disable States [1].....	8
Figure 2-5 Use of Control and Uncontrolled Ports [1]	9
Figure 2-6 Authenticator, Supplicant, and Authentication Server Roles [1].....	9
Figure 2-7 Systems Adopting both Authenticator and Supplicant Roles [1].....	10
Figure 3-1 PPP Frame Format [9].....	16
Figure 4-1 SSL/TLS Protocol Stack [10].....	23
Figure 4-2 PEAP Architecture Components [12]	28
Figure 4-3 PEAP Handshake [12].....	28
Figure 5-1 Example of VPN Connectivity [17].....	34

List of Tables

Table 2-1 802.3/Ethernet EAPOL Frame Definitions	11
Table 2-2 Token Ring/FDDI EAPOL Frame Definitions.....	12
Table 2-3 Key Descriptor Definitions.....	12
Table 3-1 EAP Types used in Request/Response Exchanges.....	19
Table 3-2 EAP Message Format [10]	20
Table 4-1 SSL Versus TLS Capabilities.....	24
Table 4-2 TTLS Field Attributes	26
Table 4-3 EAP Type Functional Comparison.....	31
Table 4-4 Detailed Comparison of EAP Methods [13]	32

1 Introduction

The NCS is part of the Department of Homeland Security Information Assurance and Infrastructure Protection (IAIP) Directorate. This Directorate analyses intelligence and information received from the NCS and other agencies involving threats to homeland security and evaluates vulnerabilities in the nation's infrastructure.

The NCS, as part of its mission, identifies new technologies and applications that enhance NS/EP communications capabilities and ensures that key NS/EP features, such as priority access, interoperability, reliability, availability, and security, are supported by emerging standards. In concert with this approach, the NCS Technology and Programs Division (N2) manages the Federal Telecommunications Standards Program in support of the NCS mission.

The purpose of this Technical Information Bulletin (TIB) is to examine the Institute of Electrical and Electronics Engineers (IEEE) 802.1X Standard for Local and Metropolitan Area Networks for Port-Based Network Access Control and its development. It discusses the definition of Extensible Authentication Protocol (EAP) and how it relates to both wired and wireless networks. It also compares the three most commonly used non-proprietary EAPs: Transport Layer Security (TLS) protocol, Tunneled Transport Layer Security (TTLS) protocol, and Protected Extensible Authentication Protocol (PEAP).

IEEE 802.1X standard defines a mechanism for controlling the access to networks via a computer's port. 802.1X was originally developed for the wired world when networking expanded beyond the physical boundaries of a building. Today, many organizations use the same facilities as their competitors. This physical access to a common facility may enable intruders to compromise information to which they do not have a legitimate right (e.g., industrial espionage). The IEEE 802.1X standard provides an added level of security and may allow the deployment of such technologies as wireless networks and virtual private networks (VPNs).

IEEE 802.1X can be used to secure existing networks and possibly add new services. Specifically, this discussion will focus on how 802.1X can support Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), 802.11 Networks, and virtual private networks (VPNs).

1.1 Ethernet and Token Ring/FDDI Networks

When 802.1X was drafted, the most popular wired networks being implemented were Ethernet, Token Ring, and FDDI. The committee decided to accommodate all of these networks. The standard specifies EAP over LAN (EAPOL) frame formats for both Ethernet and Token Ring/FDDI networks. Token Ring and FDDI networks use the same token passing mechanism; however, Token Ring has a maximum speed of 16 Mbps, whereas FDDI's is 100 Mbps. **Figures 1-1, 1-2, and 1-3** depict typical Ethernet, Token Ring, and FDDI networks.

Defining frame formats for all the popular networks allows organizations to begin implementing the security features of the 802.1X, to immediately protect their networks. Furthermore, one of the goals of 802.1X is to allow for growth in the development of

future networking technologies. The EAPOL-Key reserved a range of values that can be used to define newly developed keys on an as-needed basis.

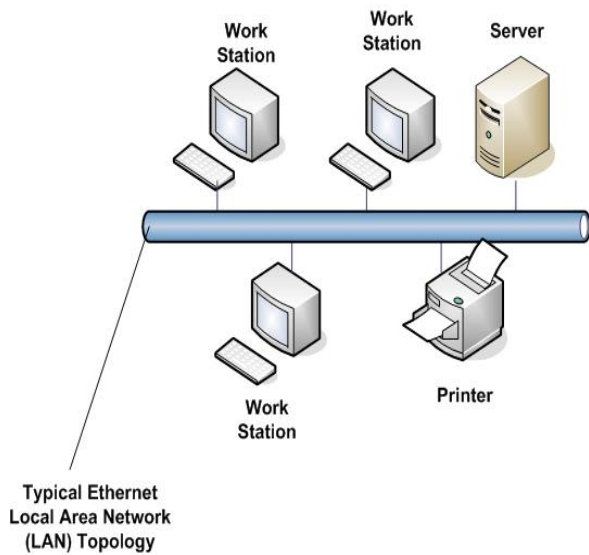


Figure 1-1 Typical Ethernet Network Topology

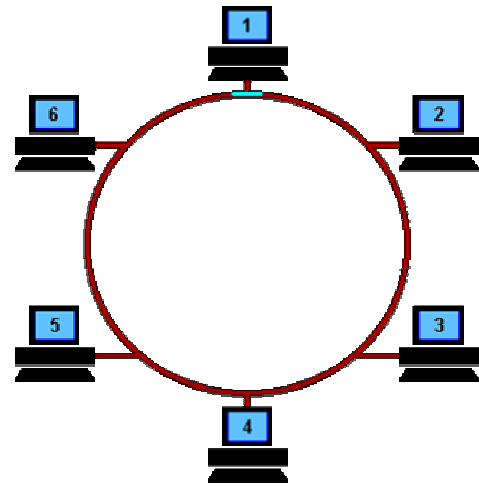


Figure 1-2 Typical Token Ring Network [14]

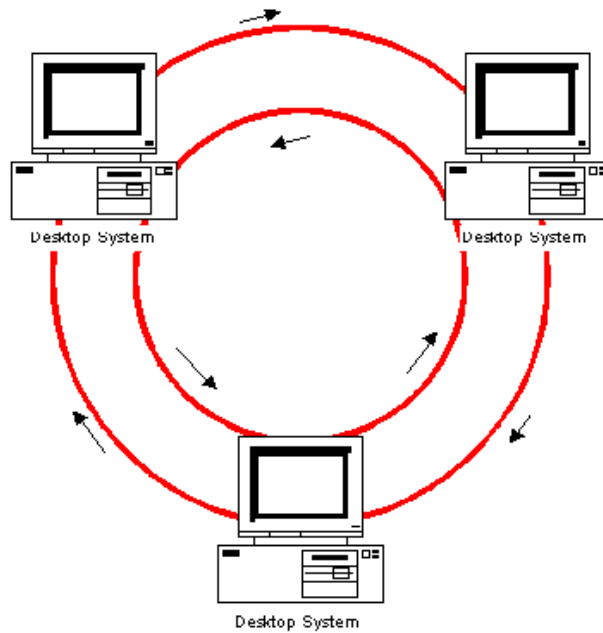


Figure 1-3 Typical FDDI Network [15]

2 Description of IEEE 802.1X

802.1X defines a mechanism for port-based network access control that makes use of the physical access characteristics of IEEE 802 LAN infrastructures in order to:

- Provide a means of authenticating and authorizing devices attached to a LAN port
- Prevent access to that port if the authentication and authorization process fails.

It is part of a family of standards for local and metropolitan area networks. The relationship between the standard and other members of the family is shown in **Figure 2-1**. (The numbers in that figure refer to IEEE standards numbers.)

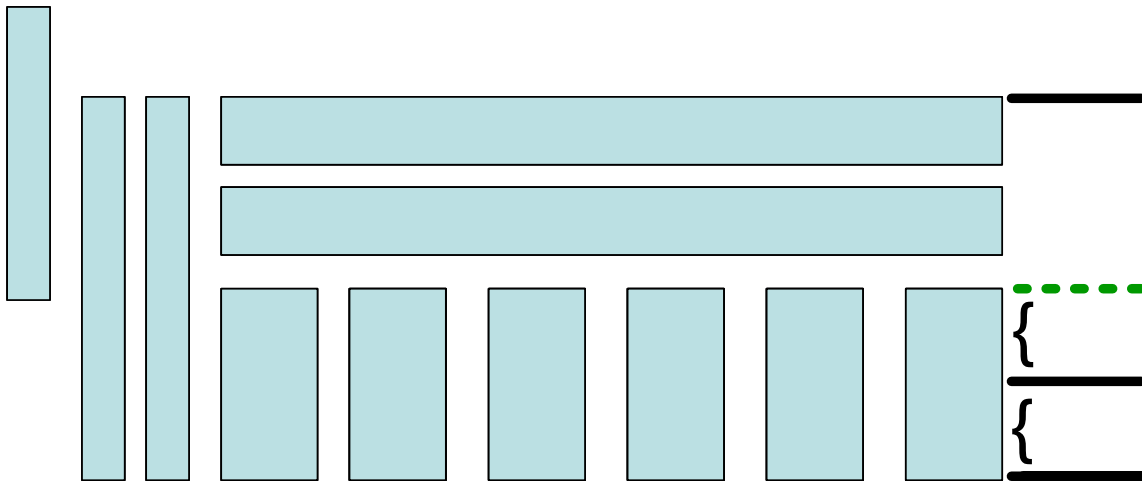


Figure 2-1 802 Family of Standards for Local Area Networks [1]

This family of standards deals with the Physical and Data Link Layers as defined by the International Organization for Standardization (ISO) Open Systems Interconnection Basic Reference Model.¹ The access standards define several types of access technologies and associated physical media, each appropriate for particular applications or system objectives.

2.1 Scope

Typically, IEEE 802 Local Area Networks (LANs) are deployed in environments that permit a wide array of devices to be physically attached to the LAN infrastructure, or permit users to attempt to remotely access the LAN through equipment already attached. Examples of such environments include corporate LANs that provide LAN connectivity in areas of a building that are accessible to the general public, and LANs that are deployed by one organization in order to offer connectivity services to other

¹ ISO/IEC 7498-1; 1994.

802.10 SECURITY

2.0 OVERVIEW & ARCHITECTURE *

802.1 MANAGEMENT

802.3 Ethernet MEDIUM ACCESS

802.4 Token Bus MEDIUM ACCESS

802.2

802.1

802.5 Token Ring MEDIUM ACCESS

organizations (e.g., as may occur in a business park or a serviced office building). In such environments, it may be necessary to restrict access to the services and data offered by the LAN to those users and devices that are permitted to make use of those services.

Port-based network access control makes use of the physical access characteristics of IEEE 802 LAN infrastructures, in order to provide a means of authenticating and authorizing devices attached to a LAN port that have point-to-point connection. A port in this context is a single point of attachment to the LAN infrastructure. The use of authentication can be desirable in certain instances, for example, the ports of Media Access Control (MAC) bridges, the ports used to attach servers or routers to the LAN infrastructure, and associations between stations and access points in IEEE 802.11 Wireless LANs.

2.2 Purpose

802.1X specifies a general protocol for the provision of port-based network access control. The standard:

- Describes the architectural framework within which the authentication takes place
- Defines the principles of operation of the access control mechanisms
- Defines the different levels of access control that are supported and the behavior of the port, with respect to the transmission and reception of frames at each level of access control within IEEE STD 802.1X
- Establishes the protocol requirements between the device that requires the authentication to occur and the attached device
- Establishes the protocol requirements between the authenticator and an authentication server
- Specifies network access control mechanisms and procedures supported by authentication and authorization protocols
- Specifies the encoding of the Protocol Data Units (PDUs) used in authentication and authorization protocol exchanges
- Establishes the requirements for managing port-based access control, identifying the managed objects and defining the management operations
- Specifies management operations available to a remote manager using the protocol and architectural description provided by the Simple Network Management Protocol (SNMP)
- Specifies the requirements for equipment conformance.

2.3 802.1X Principles of Operation

This section describes the architectural framework of port-based access control and the major functions associated with the architectural framework. It provides the relationship between the access control function and the operation of the device(s) within which it is deployed.

2.3.1 Systems, Ports, and System Roles

System ports are the attachment points between a computer or system and a LAN. They can be dedicated, physical connections or logical connections, such as those used in 802.11 wireless LANs. Port-based network access control allows the operation of a system's port(s) to be controlled in order to ensure that access to its services is permitted only by authorized systems.

The port-based access control protocol is defined as the Port Access Entity (PAE). The PAE is able to adopt one of two distinct roles within an access control interaction:

- Authenticator - The port that needs to enforce authentication before allowing access to services that are accessible via that port adopts the authenticator role. A state machine diagram for the authenticator PAE is contained in Appendix C.
- Supplicant - The port that needs to access the services offered by the authenticator's system adopts the supplicant role. A state machine diagram for the supplicant PAE is contained in Appendix C.

In some cases, the PAE is implemented in a separate server:

- Authentication Server - Performs the authentication function necessary to check the credentials of the supplicant on behalf of the authenticator and indicates whether the supplicant is authorized to access the authenticator's services.

All three roles are necessary to complete an authentication exchange. A given system can be capable of adopting one or more of these roles; for example, an authenticator and an authentication server can be collocated within the same system, allowing that system to perform the authentication function, without the need for communication with an external server. Similarly, a port can adopt the supplicant role in some authentication exchanges and the authenticator role in others. An example of the latter may be found in a bridged LAN, where a new bridge added to the LAN must be successfully authenticated by the port of the bridge via which it connects to the LAN, before it can authenticate other systems that attach to its ports.

2.3.2 Port Access Entity

The PAE executes and uses the algorithms and protocols associated with the authentication mechanisms. In the supplicant role, the PAE responds to requests from an authenticator for information that will establish its credentials. The PAE that performs the supplicant role in an authentication exchange is known as the supplicant PAE. In the authenticator role, the PAE communicates with the supplicant, and submits the supplicant information to a suitable authentication server for checking the credentials and for establishing the authorization state. The authenticator PAE controls the authorized/unauthorized state of its controlled port.

2.3.3 Controlled and Uncontrolled Access

Port-based authentication systems have two logical points of access to the LAN, as shown in **Figure 2-2**. One point of access allows the uncontrolled exchange of PDUs between the system and the LAN, regardless of the authorization state (the uncontrolled port). The other point of access allows the exchange of PDUs only if the current state of

the port is authorized (the controlled port). The uncontrolled and controlled ports are considered part of the same point of attachment to the LAN; any frame received on the physical port is made available at both the controlled and uncontrolled ports, subject to the authorization state associated with the controlled port.

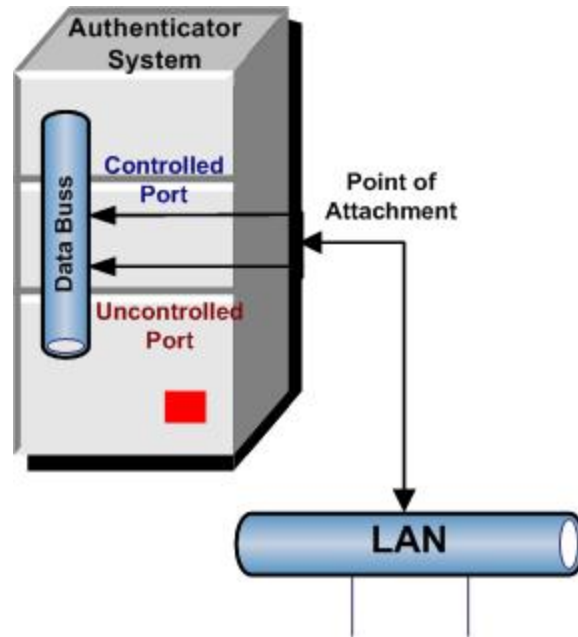


Figure 2-2 Uncontrolled and Controlled Ports [1]

Arrows are used in **Figure 2-2** and subsequent diagrams to indicate the connectivity that is available in the various configurations illustrated. For example, in **Figure 2-2**, the upward pointing arrows indicate that incoming frames can reach users attached to both the controlled and uncontrolled ports; the downward pointing arrows indicate that outbound frames from either the controlled or uncontrolled port can reach the LAN.

Any physical or logical port that can make a one-to-one connection to a supplicant system can provide the point of attachment to the LAN. For example, a single LAN MAC in a switched LAN infrastructure can provide the point of attachment. In LAN environments where the MAC method allows a one-to-many relationship between an authenticator and a supplicant, the creation of a distinct association between a single supplicant and a single authenticator is a necessary precondition for the access control mechanisms described in this standard to function. IEEE Standard 802.11, “Wireless LAN Medium Access Control (MAC) Sublayer and Physical Layer Specification,” is an example of such a relationship between a station and an access point.

Figure 2-3 illustrates the effect of the AuthControlledPortStatus associated with the controlled port. That status is represented as a switch that can be turned on or off, thus allowing or preventing the flow of PDUs via that port. Two systems are illustrated, each with a single port. In Authenticator System 1, the AuthControlledPortStatus associated with the controlled port is unauthorized and is therefore disabled (the switch is turned off); in Authenticator System 2, the AuthControlledPortStatus is authorized and is therefore enabled (the switch is turned on).

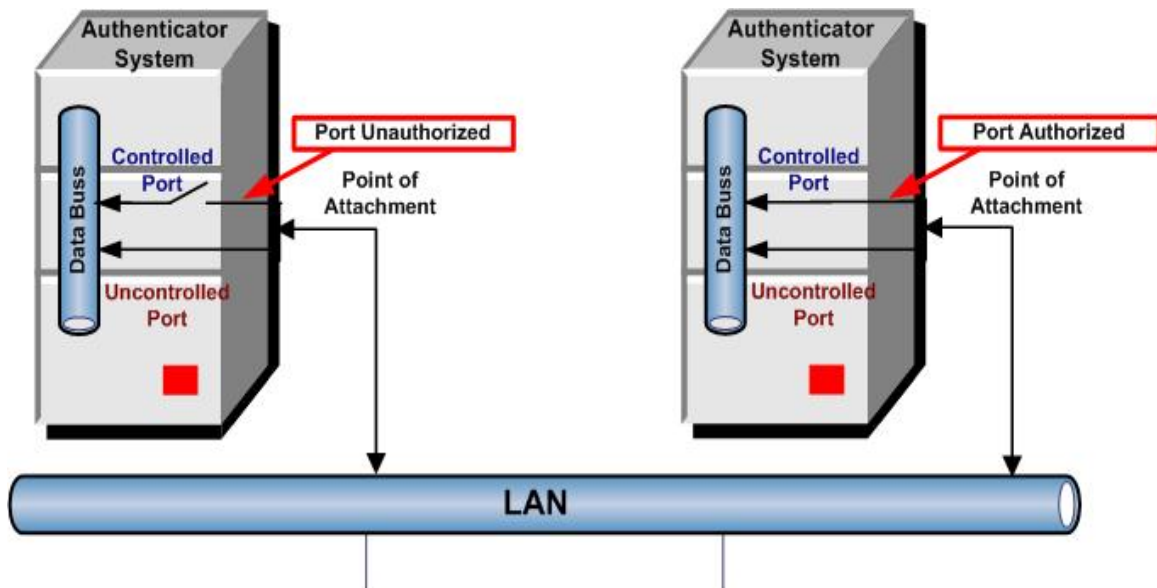


Figure 2-3 Effect of Authorization State on Controlled Ports [1]

In addition to the `AuthControlledPortStatus`, an `AuthControlledPortControl` parameter associated with the port allows administrative control over the port's authorization status. This parameter can take the values `ForceUnauthorized`, `Auto`, and `ForceAuthorized`; its default value is "Auto". The relationship between the `AuthControlledPortStatus` and `AuthControlledPortControl` parameters is as follows:

- An `AuthControlledPortControl` value of "ForceUnauthorized" forces the authenticator PAE state machine to set the value of `AuthControlledPortStatus` as unauthorized; i.e., the controlled port is unauthorized unconditionally.
- An `AuthControlledPortControl` value of "ForceAuthorized" forces the authenticator PAE state machine to set the value of `AuthControlledPortStatus` as authorized; i.e., the controlled port is authorized unconditionally.
- An `AuthControlledPortControl` value of "Auto" allows the authenticator PAE state machine to control the value of `AuthControlledPortStatus` to reflect the outcome of the authentication exchanges between the supplicant PAE, the authenticator PAE, and the authentication server.

In all three cases, the value of `AuthControlledPortStatus` directly reflects the value of the `portStatus` variable maintained by the authenticator PAE state machine.

The value of the "AuthControlledPortControl" parameter for every port of a system can be overridden by means of the "SystemAuthControl" parameter. This parameter can take the values `enabled` and `disabled`; its default value is `disabled`. If `SystemAuthControl` is set to `enabled`, then authentication is enabled for the system, and each port's authorization status is controlled in accordance with the value of the port's `AuthControlledPortControl` parameter. If `SystemAuthControl` is set to `disabled`, then all ports behave as if their `AuthControlledPortControl` parameter is set to `Force Authorized`. In effect, setting the `SystemAuthControl` parameter to `disabled` causes authentication to be disabled on all ports and forces all ports to be authorized.

Any access to the LAN is subject to the current administrative and operational state of the MAC (or logical MAC) associated with the port, in addition to AuthControlledPortStatus. If the MAC is physically or administratively inoperable, then no protocol exchanges of any kind can take place using that MAC on either the controlled or the uncontrolled port. **Figure 2-4** provides a graphic representation. In Authentication System 1, both the controlled and uncontrolled ports are able to access the LAN, because the controlled port is authorized, and the MAC providing the point of attachment to the LAN is operable. In Authentication System 2, neither the controlled nor the uncontrolled port can access the LAN, because the MAC providing the point of attachment to the LAN is inoperable. The inoperable state of the MAC has also caused the authenticator PAE to transition the controlled port to the unauthorized state, as shown in the diagram.

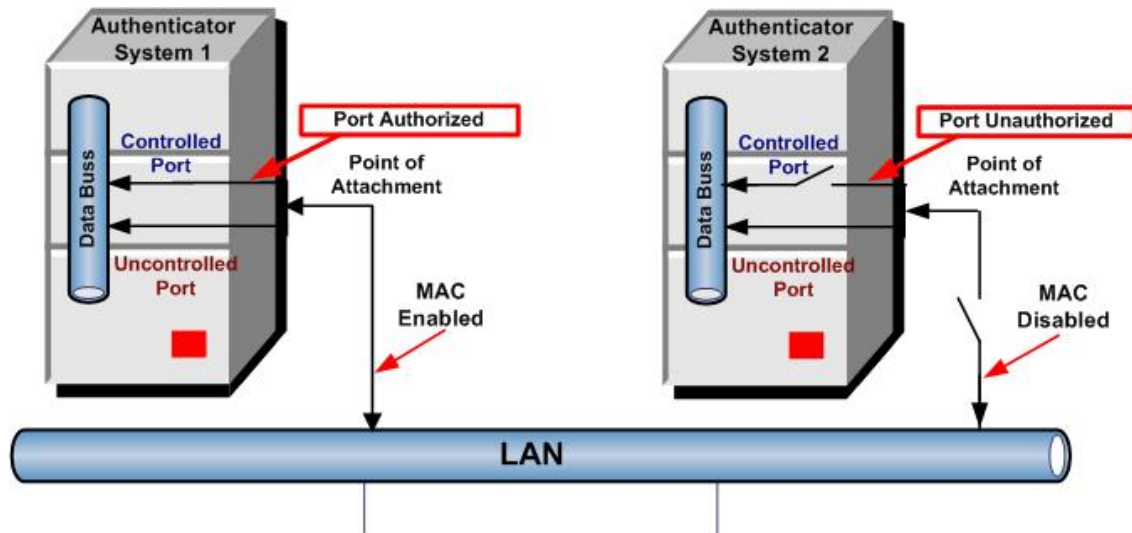


Figure 2-4 Effect of MAC Enable/Disable States [1]

The authenticator PAE uses the uncontrolled port for exchanging protocol information with the supplicant.

Protocol exchanges between the authenticator PAE and the authentication server (if the server is not collocated with the authenticator PAE) can be conducted via one or more of the system's controlled or uncontrolled ports.

Most protocol exchanges conducted by other functions of the system will make use of one or more of the system's controlled ports. However, a given protocol may need to bypass the authorization function and make use of the uncontrolled port. **Figure 2-5** shows the uses of the controlled and uncontrolled ports and the ability of the authenticator PAE to change the authorization state of its controlled port, depending on the outcome of an authentication exchange. The figure also illustrates a protocol entity that requires the use of the uncontrolled port in order to conduct its protocol exchanges.

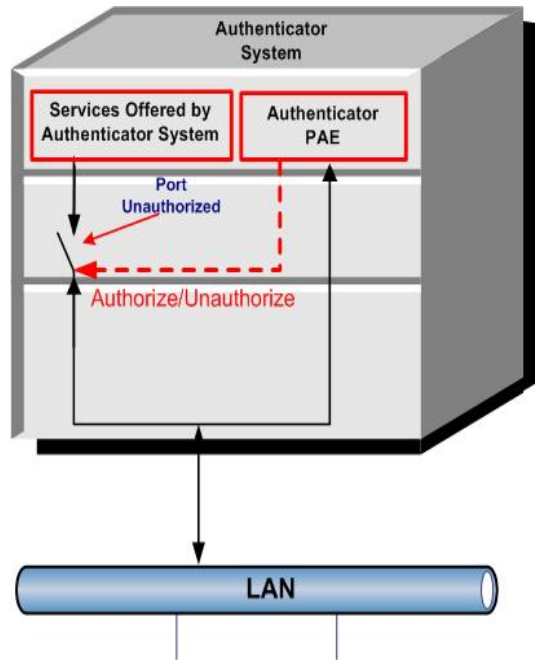


Figure 2-5 Use of Control and Uncontrolled Ports [1]

Figure 2-6 illustrates the relationship among the supplicant, the authenticator, and the authentication server, as well as the exchange of information among them. In this illustration, the authenticator’s controlled port is in the unauthorized state and is therefore disabled in terms of access to the services offered by the authenticator’s system. The authenticator PAE uses the uncontrolled port to communicate with the supplicant PAE, via Extensible Authentication Protocol Encapsulation over LANs (EAPOL) protocol exchanges and communicates with the authentication server using EAP.

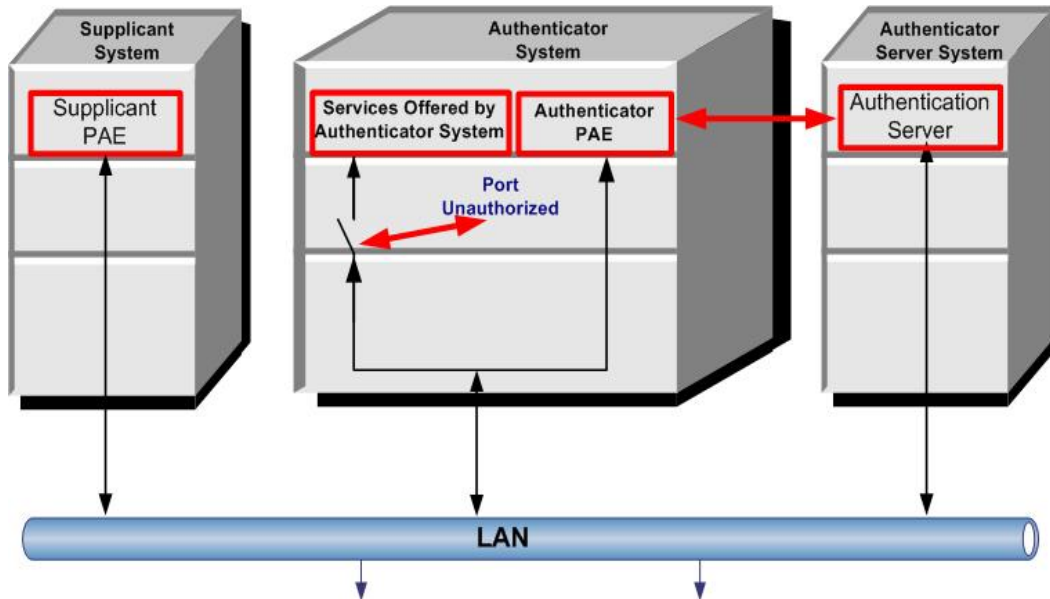


Figure 2-6 Authenticator, Supplicant, and Authentication Server Roles [1]

The communication between the authenticator and the authentication server may use the services of a LAN or some other communication channel. If the authentication server is collocated with the authenticator, EAP protocol exchanges between these two entities are unnecessary.

Figure 2-7 illustrates a situation in which the PAEs associated with the two systems, A and B, are able to adopt either the supplicant or the authenticator roles, as necessary. In order for System A to make use of System B's services, System A's PAE must adopt the supplicant role, and System B's PAE must adopt the authenticator role. For System B to make use of System A's services, the roles are reversed. The authentication server function is shown as residing in two distinct systems in this example, although this need not be the case.

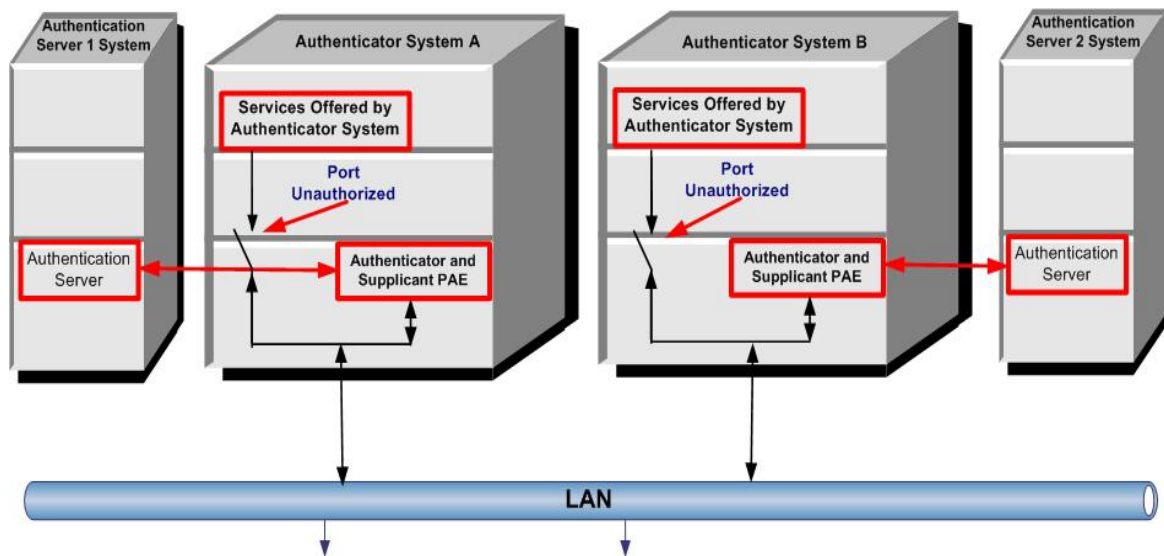


Figure 2-7. Systems Adopting Both Authenticator and Supplicant Roles [1]

Figure 2-7 Systems Adopting both Authenticator and Supplicant Roles [1]

2.4 EAP Encapsulation over LANs (EAPOL)

This section presents encapsulation techniques used to carry EAP packets between supplicant PAEs and authenticator PAEs in a LAN environment. The encapsulation is known as EAP over LANs, or EAPOL. At present, EAPOL encapsulations are described for 802.3/Ethernet and FDDI MACs and 802.5/Token Ring MACs. The EAPOL encapsulation used with 802.3/Ethernet MACs can be applied to other LAN technologies that share the same basic frame format as Ethernet (for example, IEEE STD 802.12 Demand Priority operating in IEEE STD 802.3 compatibility mode). Similarly, the EAP encapsulation used with Token Ring/Fiber Distributed Data Interface (FDDI) MACs can be applied to other LAN technologies that share the same basic frame format as IEEE STD 802.5 Token Ring (for example, FDDI or IEEE STD 802.12, "Demand Priority Access Method, Physical Layer and Repeater Specification," operating in IEEE STD 802.5 compatibility mode). A Key Descriptor identifies the encryption/decryption algorithm used for encapsulation of the EAP packets. The RC4 is the only identified algorithm in the 802.1X standard.

2.4.1 Transmission and Representation of Octets

All EAPOL PDUs consist of an integral number of octets, numbered starting from 1 and increasing in the order that they are put into a MAC frame. The bits in each octet are numbered from 1 to 8, where 1 is the low-order bit. When consecutive octets are used to represent a binary number, the lower numbered octet contains the more significant bits of the binary number.

When the encoding of an EAPOL PDU is represented in a diagram, the following representations are used:

- Octet 1 is shown at the top of the page, higher numbered octets are shown below
- Where more than one octet appears on a given line, octets are shown with the lowest numbered octet to the left and higher numbered octets to the right.
- Within an octet, bits are shown with bit 8 to the left and bit 1 to the right.

2.4.2 EAPOL Frame Format for 802.3/Ethernet

The Ethernet EAPOL frame format is defined in **Table 2-1**.

Table 2-1 802.3/Ethernet EAPOL Frame Definitions

Field Identifier	Description
PAE Ethernet Type (Octets 1-2)	2 octets in length and contains the Ethernet type assigned for use by the PAE.
Protocol Version (Octet 3)	1 octet in length. Its value identifies the version of EAPOL protocol supported by the sender of the EAPOL frame.
Packet Type (Octet 4)	1 octet in length. Its value determines the type of packet being transmitted. The following types are defined: <ul style="list-style-type: none">• Packet• EAPOL-Start• Logoff• EAPOL-Key• EAPOL-Encapsulated-ASF-Alert.
Packet Body Length (Octets 5-6)	2 octets in length. The value of this field defines the length in octets of the packet body field.
Packet Body (Octets 7-N)	Variable length. Identifies the number of packets in the field.

2.4.3 EAPOL Frame Format for Token Ring/FDDI

The Token Ring/FDDI EAPOL frame format is defined in **Table 2-2**.

Table 2-2 Token Ring/FDDI EAPOL Frame Definitions

Field Identifier	Description
SNAP-Encoded Ethernet Type (Octets 1-8)	8 octets in length and contains the SNAP-encoded Ethernet Type
Protocol Version (Octet 9)	1 octet in length. Its value identifies the version of EAPOL protocol supported by the sender of the EAPOL frame.
Packet Type (Octet 10)	1 octet in length. Its value determines the transmitted packet type. The following types are defined: <ul style="list-style-type: none"> • A value of 0000 0000 indicates it is an Packet, • A value of 0000 0001 indicates it is an EAPOL-Start. • A value of 0000 0010 indicates it is an EAPOL-Logoff • A value of 0000 0011 indicates it is an EAPOL-Key • A value of 0000 0100 indicates it is an EAPOL-Encapsulated-ASF-Alert
Packet Body Length (Octets 11-12)	2 octets in length. The value of this field defines the length in octets of the packet body field. A value of 0 indicates not Packet Body Field is present
Packet Body (Octets 13-N)	Variable length. Identifies the number of packets in the field.

2.4.4 Key Descriptor

Key algorithms are used to encapsulate the EAPOL packets. Key Descriptor types and their associated values are shown in **Table 2-3**.

Table 2-3 Key Descriptor Definitions

Descriptor Format	Definition
Descriptor Type (Octet 1)	A value of 1 in the descriptor type field indicates that the key descriptor is an RC4 descriptor. All other possible values of descriptor type are reserved for future standardization.
Key Length (Octets 2-3)	This field is two octets in length, taken to represent an unsigned binary number. The value defines the length of the key in octets. For example, a value of 5 in this field indicates a 40-bit key.
Replay Counter (Octets 4-11)	This field is 8 octets in length, taken to represent an unsigned binary number. It carries a counter value used to detect and prevent replay of key messages.
Key IV (Octets 12-27)	This field carries a 16-octet initialization vector value, consisting of 128 bits of random data.

Descriptor Format	Definition
Key Index (Octet 28)	This field is one octet in length, taken to represent a 7-bit unsigned binary number and a flag. The value is generated by the authenticator specifying the key and is used as a key index number if multiple keys are supported. The index number is carried in bits 1 through 7 and can carry an integer in the range 0-127. Bit 8 is a flag bit. If bit 8 is set to 1, the key is a unicast key; if bit 8 is set to 0, the key is a broadcast key.
Key Signature (Octets 29-44)	This field is 16 octets in length. It is a signature of all of the EAPOL packet fields, from and including the EAPOL protocol version field, to and including the encrypted key field, with the signature set to 0.
Key (Octet 45-Packet Body Length)	This field is optional. If it is not present, the supplicant uses the peer key, generated as part of the EAP authentication process, as the key material for this message. If the key is longer than the key length specified in the message, then only the first N bytes are used.

The RC4 Key Descriptor characteristics are as follows:

- The replay counter field carries an NTP time value (see IETF RFC1305 [2]).
- The key IV field carries a random number used to generate an RC47 encryption key.
- A signature type of Hashed Message Authentication Code (HMAC) - MD5 - is used to generate the key signature (see IETF RFC 2104 [3]). The key used for the signature is the server key generated by the EAP authentication (e.g., as defined in IETF RFC 2716 [4]).
- RC4² is used to encrypt the key field. The RC4 encryption key is generated by concatenating the key IV and the session key generated by the EAP authentication process (e.g., as defined in IETF RFC 2716). The key material is then encrypted according to the method specified by encrypt type (e.g., RC4 encrypted using the RC4 key).

² An RSA encryption algorithm.

3 Extensible Authentication Protocol (EAP)

Extensible Authentication Protocol (EAP) is an authentication framework which supports multiple authentication methods. EAP typically runs directly over data link layers, such as Point-to-Point Protocol (PPP) or IEEE 802, without requiring Internet Protocol (IP). EAP provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees. Fragmentation³ is not supported within EAP itself; however, individual EAP methods may support this.

EAP may be used on dedicated links as well as switched circuits, and wired as well as wireless links. To date, EAP has been implemented with hosts and routers that connect via switched circuits or dial-up lines using PPP [7]. It has also been implemented with switches and access points using IEEE 802 [802]. EAP encapsulation on IEEE 802 wired media is described in IEEE-802.1X [1] and encapsulation on IEEE wireless LANs in IEEE-802.11i [8]. Currently, there are over 60 EAPs available on the market.

One of the advantages of the EAP architecture is its flexibility. EAP is used to select a specific authentication mechanism, typically after the authenticator requests more information in order to determine the specific authentication method to be used. Rather than requiring the authenticator to be updated to support each new authentication method, EAP permits the use of a back end authentication server which may implement some or all authentication methods, with the authenticator acting as a pass-through for some or all methods and peers.

3.1 Point to Point Protocol

The Point-to-Point Protocol is a Layer 2 or Data Link Layer protocol that allows two peer devices e.g., two host computers, or a host computer and a bridge or router, to transport packets over a simple link. PPP is commonly used to support Transmission Control Protocol/Internet Protocol (TCP/IP) traffic between an asynchronous personal computer (PC) and an access router for Internet access over a dial-up serial link. This is generally the way users connect across the Public Switched Network (PSN) from their PC to an Internet service provider (ISP). PPP is a connection-oriented protocol that encapsulates packet data using a variation on the HDLC protocol. It supports full duplex transmission, both synchronous and asynchronous. PPP includes error detection and data protection features. PPP supports these functions by providing an extensible Link Control Protocol (LCP) and a family of Network Control Protocols (NCPs) to negotiate optional configuration parameters and facilities. PPP provides a common solution for easy connection of a wide variety of hosts, bridges, and routers.

3.1.1 PPP Elements

PPP provides a method for transmitting datagrams⁴ over serial point-to-point links. It contains three main components:

³ In messaging, it is the process in which an IP datagram is broken into smaller pieces to fit the requirements of a given physical network. The reverse process is termed “reassembly.”

⁴ A transmission method in which sections of the message are transmitted in scattered order and the correct order is reestablished by the receiving workstation.

- A method for encapsulating datagrams over serial links. PPP uses the High-Level Data Link Control (HDLC) protocol as a basis for encapsulating datagrams over point-to-point links.
- An extensible LCP to establish, configure, and test the data link connection.
- A family of NCPs for establishing and configuring different network layer protocols. PPP is designed to allow the simultaneous use of multiple network layer protocols.

To establish communications over a point-to-point link, the originating PPP first sends LCP frames to configure and (optionally) test the data link. After the link has been established and optional facilities have been negotiated as needed by the LCP, the originating PPP sends NCP frames to choose and configure one or more network layer protocols. When each of the chosen network layer protocols has been configured, packets from each network layer protocol can be sent over the link. The link will remain configured for communications until explicit LCP or NCP frames close the link or some external event occurs (for example, an inactivity timer expires or a user intervenes).

3.1.2 Physical Layer Requirements

PPP is capable of operating across any Data Terminal Equipment/Data Communications Equipment (DTE/DCE) interface. Examples include Electronic Industries Alliance/Telecommunications Industry Association (EIA/TIA) EIA/TIA-232-C (formerly RS-232-C), EIA/TIA-422 (formerly RS-422), EIA/TIA-423 (formerly RS-423), and International Telecommunication Union Telecommunication Standardization Sector (ITU-T) (formerly CCITT) V.35. The only absolute requirement imposed by PPP is the provision of a duplex circuit, either dedicated or switched, that can operate in either an asynchronous or synchronous bit-serial mode, transparent to PPP Link Layer frames. PPP does not impose any restrictions regarding transmission rate other than those imposed by the particular DTE/DCE interface in use. Link Layer PPP uses the principles, terminology, and frame structure of the International Organization for Standardization (ISO) HDLC procedures (ISO 3309-1979), as modified by ISO 3309:1984/PDAD1 “Addendum 1: Start/Stop Transmission.” ISO 3309-1979 specifies the High Level Data Link Control (HDLC) frame structure for use in synchronous environments. ISO 3309:1984/PDAD1 specifies proposed modifications to ISO 3309-1979 to allow its use in asynchronous environments. The PPP control procedures use the definitions and control field encodings standardized in ISO 4335-1979 and ISO 4335-1979/Addendum 1-1979. The PPP frame format appears in **Figure 3-1**.

FIELD LENGTH, IN BYTES	1	1	1	2	VARIABLE	2 OR 4
	FLAG	ADDRESS	CONTROL	PROTOCOL	DATA	FCS

Figure 3-1 PPP Frame Format [9]

The following descriptions summarize the PPP frame fields illustrated in **Figure 3-1**:

- Flag – A single byte that indicates the beginning or end of a frame. The flag field consists of the binary sequence 01111110.
- Address – A single byte that contains the binary sequence 11111111, the standard broadcast address. PPP does not assign individual station addresses.
- Control – A single byte that contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame. A connectionless link service similar to that of Logical Link Control (LLC) Type 1 is provided.
- Protocol – Two bytes that identify the protocol encapsulated in the information field of the frame.
- Data – Zero or more bytes that contain the datagram for the protocol specified in the protocol field. The end of the information field is found by locating the closing flag sequence and allowing 2 bytes for the FCS field. The default maximum length of the information field is 1,500 bytes. By prior agreement, consenting PPP implementations can use other values for the maximum information field length.
- Frame check sequence (FCS) – Normally 16 bits (2 bytes). By prior agreement, consenting PPP implementations can use a 32-bit (4-byte) FCS for improved error detection.

The LCP can negotiate modifications to the standard PPP frame structure. Modified frames, however, always will be clearly distinguishable from standard frames.

3.1.3 PPP Link-Control Protocol

The PPP LCP provides a method of establishing, configuring, maintaining, and terminating the point-to-point connection. LCP goes through four distinct phases:

1) Link establishment and configuration negotiation occur. Before any network layer datagrams (for example, IP) can be exchanged, LCP first must open the connection and negotiate configuration parameters. This phase is complete when a configuration-acknowledgment frame has been both sent and received.

2) LCP allows an optional link quality determination phase. In this phase, the link is tested to determine whether the link quality is sufficient to bring up network layer protocols. LCP can delay transmission of network layer protocol information until this phase is complete.

3) Network layer protocol configuration negotiation occurs. After LCP has finished the link quality determination phase, network layer protocols can be configured separately by the appropriate NCP and can be brought up and taken down at any time. If LCP closes the link, it informs the network layer protocols so that they can take appropriate action.

4) Link termination occurs. LCP can terminate the link at any time. This usually is done at the request of a user but can happen because of a physical event, such as the loss of carrier or the expiration of an idle-period timer.

Three classes of LCP frames exist. Link-establishment frames are used to establish and configure a link. Link-termination frames are used to terminate a link, and link-

maintenance frames are used to manage and debug a link. These frames are used to accomplish the work of each of the LCP phases.

3.2 Point-to-Point Protocol Extensible Authorization Protocol

The PPP Extensible Authentication Protocol (EAP) is a general protocol for PPP authentication which supports multiple authentication mechanisms. EAP does not select a specific authentication mechanism at Link Control Phase, but rather postpones this decision until the Authentication Phase. This allows the authenticator to request more information before determining the specific authentication mechanism. This also permits the use of a back end server which actually implements the various mechanisms, while the PPP authenticator merely passes through the authentication exchange.

3.2.1 Authorization Exchange

EAP uses a set of messages to initiate and to complete the authentication exchange. These are used with all upper-layer authentication methods. EAP also allows two parties to exchange information that is specific to the authentication method they want to use. The content of these authentication-specific methods is not defined in EAP. In fact, they can be completely proprietary authentication methods or newly invented ones. EAP's ability to handle part of the communication in a standardized way and part in a specific way is the key to its extensibility. These authentication-specific messages are referred to as "middle messages" because they occur after the initiation and before the completion of the authentication exchange.

Numerous middle messages can be exchanged before the authentication is completed. EAP is extensible because the details of these special messages are left to other RFCs to fill in. For example, RFC 2246, "TLS Protocol Version 1.0," January 1999 (as updated by RFC 3546, "TLS Extension,") states how to use Transport Layer Security (TLS) over EAP; another (draft) states how to use Tunneled TLS (TTLS) over EAP[21], and so on. If a new method is invented in the future, a new draft can be written called, "mymethod over EAP." If it becomes popular, users can implement it on existing systems.

RFC2284 [5], "PPP EAP," specifies that four types of messages can be sent:

- Request – After the link establishment phase is complete, the authenticator sends one or more requests to authenticate the peer. The request has a type field to indicate what is being requested. Examples of request types include Identity, MD5-challenge, One-Time Passwords, Generic Token Card, etc. The MD5-challenge type corresponds closely to the RFC 2759, "PPP Challenge Handshake Authentication Protocol (CHAP) [18] authentication protocol. Typically, the authenticator will send an initial identity request followed by one or more requests for authentication information. However, an initial identity request is not required, and may be bypassed in cases where the identity is presumed (leased lines, dedicated dial-ups, etc.).
- Response – The peer sends a response packet in reply to each request. As with the request packet, the response packet contains a type field which corresponds to the type field of the request.
- Success – Sent by the authenticator to indicate access is granted.

- Failure – Sent by the authenticator to indicate access is refused.

These messages are described in terms of the authenticator. However, in the IEEE 802.1X scenario, the authenticator forwards the messages on to the authentication server, most likely using RADIUS. In this case it is the authentication server that generates request, success, and/or failure messages and the authenticator just relays them to the supplicant.

3.2.2 Request and Response Messages

Request and response messages are subdivided using the EAP type field. The type field indicates what information is being carried in the EAP message. The first six message types are defined in the RFC 2284 and shown in **Table 3-1**.

Table 3-1 EAP Types used in Request/Response Exchanges

EAP Type	Name	Description
1	Identity	Used to query the identity of the peer.
2	Notification	Optionally used to convey a displayable message from the authenticator to the peer.
3	NAK (response only)	Sent in reply to a request where the desired authentication type is unacceptable. Only valid in response messages.
4	MD-5 Challenge	The request contains a “challenge” message to the peer.
5	One-Time Password (OTP)	The request contains a displayable message containing an OTP challenge.
6	Generic Token Card	The request contains an ASCII text message and the reply contains the token card information necessary for the authentication.

Others are reserved for specific authentication methods. The most important predefined type is identity (type value 1). Typically, this is used as part of the EAP introduction phase: the message Request/Identity is sent by the authenticator to a new supplicant. The supplicant replies with the message Response/Identity containing its user name or some other identifier that will be understood by the authentication server.

Type numbers higher than 6 are not defined by RFC 2284. They are unique and issued by the Internet Assigned Numbers Authority (IANA) for each new authentication method that is introduced. Some are even issued for vendor-proprietary methods. The type number for TLS, for example, is 13, which means that all Request and Response messages with this type field contain information that is specific to the TLS upper-layer authentication method.

The use of the type field is a somewhat inconsistent. For the most part, it indicates the authentication method. But in a few cases, it defines a special-purpose message. For example, a message with a type value of 2 is called a notification message and is used to send user-displayable text, such as “Please enter your password.” The message is

intended to appear on the screen of the user's system. A message with a type value of 3 is called a Negative AcKnowledgment (NAK) and is used when a request is made for an authentication method that is not supported. If an EAP request with type TLS is sent to a peer that doesn't support TLS, it can respond with a type field of NAK.

Type value 1, identity, could be considered a special-purpose message or it could be considered a very simple authentication method. Under IEEE 802.1X, this request is often the first thing sent and the supplicant will reply with a response message giving its identity information. The simplest authentication exchange would be:

- Identity request (from authenticator)
- Identity response (from supplicant)
- Success (from authenticator).

Here the device has been “authenticated” on pure trust: “I choose to believe that you are who you say with no proof.” Or perhaps proof is available by some other means. For example, the identity might be generated by a smart card that changes every second, synchronized to the authentication server. This type of null authentication can be used with simple wireless LAN networks that have preloaded secret keys (called preshared keys) and then rely on the encryption to prevent unwanted communications.

Because the Identity exchange can be considered a complete authentication method by itself, an identity exchange followed by another method such as TLS, is in reality two authentication methods running in sequence. This concept of serial authentication has been generalized in RFC 3748, “Extensible Authentication Protocol,” [19], which simply lists the Identity message as a basic authentication method and allows the running of as many authentication methods in sequence as desired prior to the final Success or Failure message. This ability to run multiple authentication methods in sequence can be exploited in new approaches that allow the client to authenticate the network before revealing its identity.

3.2.2.1 EAP Packet Message Formats

EAP messages have a similar basic format as shown in **Table 3-2**.

Table 3-2 EAP Message Format [10]

Field Identifier	No. of Octets	Description
Code - Octet 1	1	Indicates the type of EAP packet message. EAP Codes are assigned as follows: <ul style="list-style-type: none"> • 1 for Request • 2 for Response • 3 for Success • 4 for Failure
Identifier - Octet 2	1	Allows matching of responses and requests
Length - Octets 3-4	2	Length of EAP packet
Data - Octets 5-N	≥0	The Data field is determined by the Code field

Success and failure packets are messages that are short and contain no data. One of these messages is used at the end of the authentication process to signal the result. Because success and failure are common across all authentication protocols, intermediate devices (such as the access point) can detect when an authentication completes, without understanding all the details of the authentication method.

The details of the authentication method are sent in the request and response messages. These have an extra field called type. The type field is essential to separate all the different authentication methods. In fact, it is the key to the extensibility of EAP. Each new authentication method is assigned a unique value so the system knows whether the request contains information relevant to TLS or PEAP.

3.3 Deploying EAP in a Corporate/Government Environment

Deploying an EAP is not a simple process for at least three reasons. First, the 802.1X standard does not specify that a particular EAP be used in any situation. Second, the National Institute of Standards and Technology (NIST) continues to study various EAPs, they do not make any recommendations, nor have they approved any EAP for Government use. Third, it requires a great deal of valuable resources for a full, enterprise-wide deployment.

The three most prevalent EAPs being used by corporations and government are Transport Layer Security (TLS), Tunneled Transport Layer Security (TTLS), and Protected Extensible Authentication Protocol (PEAP). These are discussed in detail in Section 4. The main requirement for any user, who wishes to deploy 802.1X, is to evaluate the organization's security needs and select the appropriate EAP. For example, a high-security organization, such as the National Security Agency, would most likely choose an EAP that supports the relatively new Advanced Encryption Standard (AES), which provides a very strong 256-bit encryption algorithm. An organization that does not need to protect data as strongly might choose an EAP like TLS that supports a 128-bit encryption algorithm. The 802.1X recommends that the EAP selected support mutual authentication. The three EAPs, TLS, TTLS and PEAP, discussed in the standard all support mutual authentication.

4 Commonly Used EAP Methods

Currently, the three most widely deployed Encapsulated Authentication Protocols are Transport Layer Security (TLS), Tunneled Transport Layer Security (TTLS), and Protected Extensible Authentication Protocol (PEAP). Each of these will be examined in the following sections.

4.1 Transport Layer Security (TLS)

Transport Layer Security (RFC 2246) is the IETF standard for Transport Layer Security (TLS). TLS is based on Netscape Communications Secure Socket Layer (SSL) technology. SSL is a protocol that uses digital cryptography to secure the communication between clients and services. The latest version is SSL version 3.0, issued in 1996. Although Netscape holds the patents for SSL, it has made the specifications and source code publicly available. Consequently, SSL has achieved wide acceptance and grown stronger, as a result of public scrutiny. TLS adds functionality to strengthen the security capability and is backward compatible with SSL. It includes mechanism that can be implemented to downwardly shift to SSL 3.0 operation when both parties do not support TLS. Because SSL and TLS implement the same fundamental technology, the protocols will often be identified by the combined acronym SSL/TLS.

Figure 4-1 illustrates where SSL/TLS fits in the Internet protocol stack. SSL/TLS interfaces with upper-layer applications that are specifically designed to work with it. There are two phases. The first phase or handshake phase, coordinates the establishment of the connection between networks. Once the handshake has been completed, the second phase establishes a dedicated communications path between the two application layers. Note that the top layer of the SSL/TLS model does not consist of a generic process/application layer. Because SSL/TLS replaces the interface that TCP presents to applications, standard TCP applications cannot communicate using SSL/TLS. A separate, secure version of the application must be written to take advantage of SSL/TLS services. The best-known example of an application that uses SSL/TLS is Secure HTTP (HTTPS).

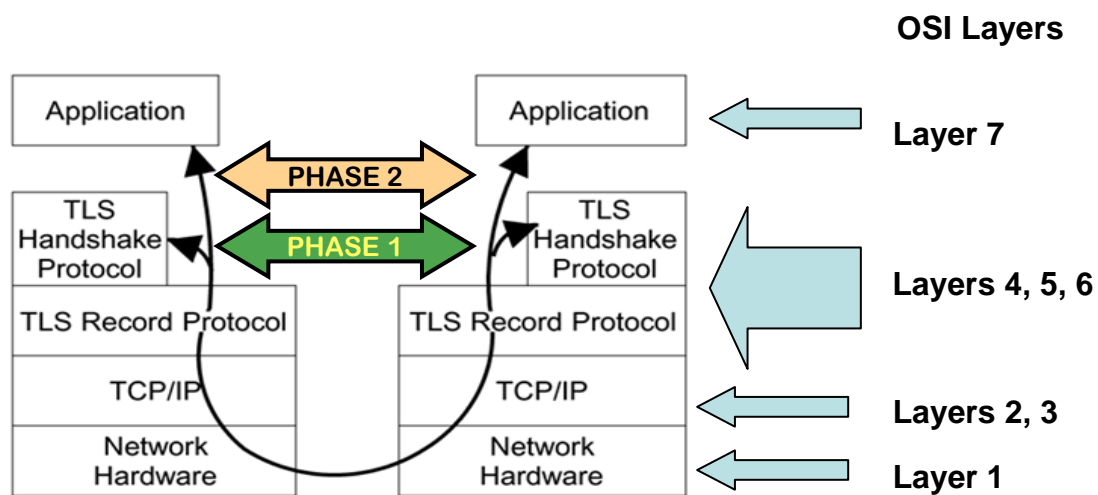


Figure 4-1 SSL/TLS Protocol Stack [10]

4.1.1 TLS Functionality

SSL supports a variety of cryptographic algorithms, and TLS supports a few additional ones. It is beyond the scope of this TIB to discuss all available ciphers. **Table 4-1** illustrates the TLS contribution to TCP/IP communication.

Table 4-1 SSL Versus TLS Capabilities

SSL Vs TLS	Authentication	Integrity	Confidentiality
SSL Ver 3.0	Clients Can Authenticate Servers	SSL uses message digest mechanism to ensure data integrity. Digital signatures are not used.	After an initial handshake, all data above TCP are encrypted using a symmetric cipher.
TLS Ver 1.0	Clients can Authenticate other clients	TLS uses message digests and digital signatures to ensure that data are not modified in transit.	TLS permits additional cryptographic ciphers not allowed in SSL

4.1.2 TLS Operation

TLS makes use of two protocols: a handshake protocol to initialize a session and a record protocol to exchange data. Before TLS can exchange data securely, a handshake process must be executed to enable the client and server to agree on ciphers, to authenticate, and to generate keys. When a client attempts to communicate with a secure server, a session must be initialized using the handshake protocol as follows:

- The client sends the server a hello message that includes the client's TLS version number, supported encryption and message digest protocols, supported key lengths, and supported key exchange methods. Included is a challenge message, consisting of randomly generated data that the server must include in an authentication response.
- The server sends the client a hello message that includes the server's TLS version number, mutually supported encryption protocols, message digest protocols, key lengths, and key exchange methods. The message includes a randomly generated challenge to the client and the server's public key certificate. If the server is required to authenticate the client, the message includes a request for the client's certificate.
- The client attempts to authenticate the server by validating the server's certificate. If the server is authenticated, the client sends a master key message to the server, which includes a 384-bit premaster⁵ secret for the session, encrypted with the server's public key. The premaster secret is based on all data that the client has received from the server to that point.

⁵ Seed material used to generate the master key for a session.

- If the server has taken the optional step of requesting authentication of the client, the client sends another message which includes data that are unique to the handshake, digitally signing the message with the client's public key. The encrypted data are sent to the server with the client's certificate.
- If the server has requested client authentication, the server attempts to authenticate the client by validating the client's certificate. The session is rejected if client authentication fails. If the client is authenticated, the server decrypts the premaster secret using the server's private key and applies an algorithm to generate the 384-bit master secret.
- The client applies the same steps to the premaster secret so that client and server derive the same master secret.
- The client and server use the master secret to generate session keys, symmetric keys that are used to encrypt and decrypt data exchanged throughout the TLS session.
- The client sends a message informing the server that all further data will be encrypted with the session key. It then sends a separate, encrypted message informing the server that the client's portion of the handshake is completed.
- The server sends a message informing the client that all further data will be encrypted with the session key. It then sends a separate, encrypted message informing the client that the server's portion of the handshake is completed.
- The handshake is complete and the TLS session is initiated.

It should be noted that TLS does not protect data at the IP and TCP protocol layers, most significantly the IP addresses of the hosts. Because IP addresses do provide some information about the organization's network structure, there are situations when everything above the network access layer should be encrypted.

4.2 Tunneled Transport Layer Security

TTLS [20] is an extension of TLS [21] and provides the benefits of strong encryption, without the complexity of mutual certificates on both the client and authentication server. Like TLS, TTLS supports mutual authentication, but only requires the authentication server to be validated to the client through a certificate exchange. It allows the client to authenticate to the authentication server using usernames and passwords and only requires a certificate for the authentication servers. TTLS simplifies rollout and maintenance and retains strong security and authentication. A TLS tunnel can be used to protect EAP messages. Existing user credential services, such as Active Directory, RADIUS, and LDAP, can be reused for 802.1X authentication. TTLS also provides backward compatibility for other authentication protocols, such as PAP, CHAP, MS-CHAP, and MS-CHAP-V2.

Unlike SSL and TLS that have single phase authentication, the TTLS authentication process is broken into two phases; Phase 1: The Handshake and Phase 2: The Tunnel.

4.2.1 Phase 1: The TTLS Handshake

In Phase 1, the TLS handshake protocol is used to authenticate the TTLS server to the client and, optionally, to authenticate the client to the TTLS server. It is initiated when the client sends an Response/Identity packet to the TTLS server. This packet specifically should not include the name of the user. However, it may include the name of the realm of a trusted provider to which TTLS packets should be forwarded, for example, “@myisp.com.” The TTLS server responds to the Response/Identity packet with a TTLS/Start packet, which is an Request with Type = TTLS, the S (Start) bit set, and no data. This indicates that the client should begin the TLS handshake by sending a ClientHello message. EAP packets are exchanged between client and TTLS server to complete the TLS handshake. Phase 1 is completed when the client and TTLS server exchange ChangeCipherSpec and Finished messages. At this point, additional information may be securely tunneled.

As part of the TLS handshake protocol, the TTLS server will send its certificate along with a chain of certificates leading to the certificate of a trusted CA. The client must be configured with the certificate of the trusted CA in order to perform the authentication. For certificate-based authentication of the client, a certificate must be issued and the client must have the private key associated with that certificate.

4.2.2 Phase 2: The TTLS Tunnel

In Phase 2, the TLS record layer is used to create a secure path-way (or tunnel⁶) for information to pass between the client and the TTLS server. This information is encapsulated in sequences of Attribute-Value Pairs (AVPS). Attributes carry the specific authentication, authorization and accounting details for the request and response. The AVP format is shown in **Table 4-2**. The Type field is one octet. The Length field is one octet, and indicates the length of this attribute including the Type, Length and Value fields. The Value field is zero or more octets and contains information specific to the attribute. The format and length of the Value field is determined by the Type and Length fields.

Table 4-2 TTLS Field Attributes

Field Identifier	Number of Octets	Description
Type Octet 1	1	Refers to the specific authentication method being utilized e.g., MS-CHAP etc.
Length Octet 2	1	Indicates the length of this attribute including the Type, Length and Value fields
Value Octets 3-N	≥0	Contains information specific to the attribute

⁶ A tunnel is an intermediary program which acts as a blind relay between two connections. It ceases to exist when both ends of the connection are closed. It provides a secure path for communications between client and servers over an inherently insecure IP-based network.

Any type of information may be exchanged during Phase 2, according to the requirements of the system. (It is expected that applications utilizing TTLS will specify what information must be exchanged and therefore which AVPs must be supported.)

The client begins the Phase 2 exchange by encoding information in a sequence of AVPs, passing this sequence to the TLS record layer for encryption, and sending the resulting data to the TTLS server. The TTLS server recovers the AVPs in clear text from the TLS record layer. If the AVP sequence includes authentication information, it forwards this information to the AAA/H server using the AAA carrier protocol. Note that the TTLS and AAA/H servers may be one and the same, in which case it simply processes the information locally. The TTLS server may respond with its own sequence of AVPs. The TTLS server passes the AVP sequence to the TLS record layer for encryption and sends the resulting data to the client. For example, the TTLS server may send key distribution information, or it may forward an authentication challenge received from the AAA/H. This process continues until the TTLS server has enough information to issue either an Success or Failure. Thus, if the AAA/H rejects the client based on forwarded authentication information, the TTLS server would issue a Failure. If the AAA/H accepts the client, the TTLS server would issue a Success.

The TTLS server distributes data connection keying information and other authorization information to the access point in the same AAA carrier protocol message that carries the Success.

4.3 Protected Extensible Authentication Protocol

Protected Extensible Authentication Protocol (PEAP) is defined in an IETF Draft that has strong backing from companies such as Microsoft and RSA. Similar to the TTLS, PEAP adds a layer of security using Transport Layer Security (TLS) on top of a type (such as TLS, MD5). This extra layer of security is used to protect the integrity of EAP authentication messages.

Three hardware and software components are required to implement PEAP security for wireless LANs, the peer, the authenticator, and the authentication server. The peer is a wireless device that connects to the PEAP-enabled wireless LAN through PEAP client software. The authenticator is an 802.1X and enabled access point or wireless switch. The authenticator acts as a middleman between the peer and the authentication server during the authentication process. The authentication server provides RADIUS, EAP, and TLS services to validate the credentials of the peer and the authenticator, and connects to the wired LAN. **Figure 4-2** shows the relationship of the three hardware components.



Figure 4-2 PEAP Architecture Components [12]

Two phases of the PEAP security mechanism must be successful for a user or a device to transfer encrypted network traffic. In phase 1, a TLS session is negotiated between the server and the authenticator using an encrypted tunnel to protect the authentication information being exchanged. As a part of this process, a key is negotiated and is used for encrypting the rest of the conversation. In phase 2, an EAP is used within the TLS session to authenticate the user of the wireless device. **Figure 4-3** graphically illustrates the two-phase process.

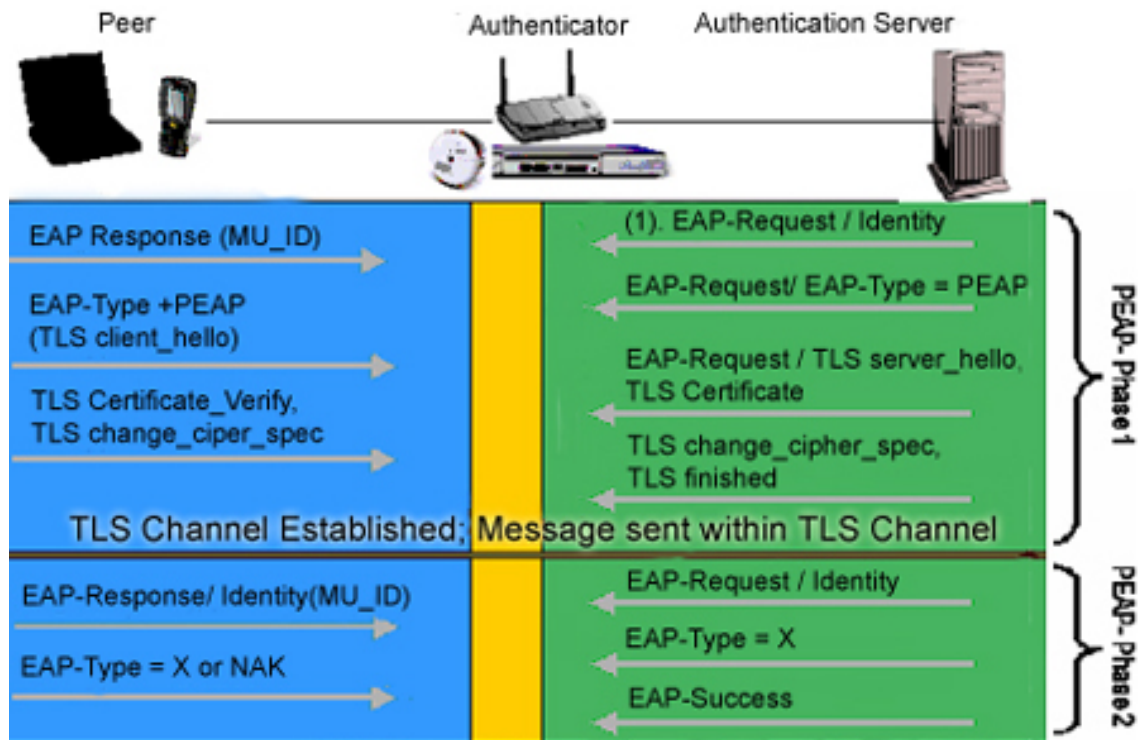


Figure 4-3 PEAP Handshake [12]

4.3.1 Phase 1: The PEAP Handshake

In Phase 1, the authenticator is authenticated to the peer using a TLS handshake. The following sequence of steps occurs in this phase:

- The MU sends a message to a back end EAP server announcing that it is connected to the AP. The message tells the server that a new connection should be initiated. In addition, the MU indicates which cryptographic algorithms it understands, so that secure messages sent between the two can be understood.
- After receiving this message, the back end EAP server responds with a new session ID, a list of algorithms (TLS cipher suites) that will be used to correspond, and a public key certificate that allows the MU to trust the AP it has used to establish the network connection. The server chooses a TLS cipher suite from those offered by the client. PEAP implementations need not support all TLS cipher suites listed in the RFC 2246. However, in order to ensure interoperability, the PEAP peers and the authenticators must be able to negotiate the following cipher suites:
 - TLS_RSA_WITH_RC4_128_MD5
 - TLS_RSA_WITH_RC4_128_SHA
- The MU verifies the signature and validity of the EAP server certificate by using a pre-loaded root certificate. The MU then responds by generating a secret key and encrypting it with the public key obtained from the server certificate. This protected information is sent back to the server.
- If the server is able to decrypt this information, the MU is authenticated. Only the server's private key is able to decrypt messages encrypted with its public key.
- After this last exchange, authentication of the AP is complete. A secure TLS session is established to protect the user authentication credentials, which will be passed in the PEAP Phase 2.
- One key difference between PEAP and EAP is that the success or failure packets are authenticated making it stronger against a forgery type of attack. A forged Failure can be used to disconnect a peer. A forged Success message can allow rogue APs access to the network.

4.3.2 Phase 2: The PEAP Tunnel

If the TLS session is successfully established in Phase 1, a Phase 2 of PEAP conversation occurs. Phase 2 consists of another complete EAP conversation within the TLS session negotiated in Phase 1. This provides an additional layer of protection, enabling strong authentication of the MU end user; the user is challenged with a suitable EAP mechanism, which includes the use of passwords, smart cards, or digital certificates.

The following sequence of steps occurs in Phase 2:

- A TLS-protected identity challenge/response exchange occurs between the AP and the MU. This prevents snooping and packet modification attacks.
- The EAP server then selects an authentication method for the MU. This could include MD5 or TLS. The MU can send a negative response (NAK) to the server and suggest

an alternative. Since the NAK is also sent within the TLS channel, it is protected from snooping or packet modification. Therefore, an attacker cannot negotiate a lower security scheme and is prevented from determining which EAP method was negotiated between the MU and the server.

- The EAP conversation encapsulated within the TLS channel continues (similar to what is described in RFC 2284) until the server sends Failure or Success to the MU.
- Once the Failure or Success message is received, the TLS channel is shut down by the MU and the server.

TLS provides a connection reestablishment mechanism allowing users to authenticate more quickly to a newer AP while roaming (compared to just using EAP). As long as the session ID is still valid, the MU and server can share old ciphers to negotiate a new handshake and keep the connection alive and secure.

4.4 Comparison of TLS, TTLS, and PEAP

TLS has many attributes that make it attractive for security-related use. It is well documented and has been analyzed quite extensively. Studies have not yet revealed significant weaknesses in the protocol itself and it is standardized by the IETF in RFC 2716. TLS authenticates peers by exchanging digital certificates. Certificates are protected on the client by a password or PIN, or stored on a smart card, depending on the implementation. One flaw in the TLS protocol noted by numerous observers is that the identity exchange proceeds in the clear before exchange of certificates; therefore, a passive attack could easily observe user names. Digital certificates are the Achilles heel of TLS. Certificate authentication of clients mandates a concurrent PKI rollout. If PKI is not already in place, the additional work involved in issuing and managing certificates is quite large. Compared to other PKI-enabled protocols, TLS may impose a greater certificate management overhead, because of the need to revoke certificates as wireless LAN access is revoked from users. Both TTLS and PEAP were developed in response to the PKI barrier in TLS. Client certificates are not ideal for user authentication for a variety of reasons. Older methods of user authentication are as secure as certificate-based authentication, but without the high management overhead. Both TTLS and PEAP were designed to use older authentication mechanisms, while retaining the strong cryptographic foundation of TLS. The structure of TTLS and PEAP are quite similar. Both are two-stage protocols that establish security in stage one and then exchange authentication in stage two. Stage one of both protocols establishes a TLS tunnel and authenticates the authentication server to the client with a certificate. (TTLS and PEAP still use certificates to authenticate the wireless network to the user, but only a few certificates are required, so it is much more manageable.) Once that secure channel has been established, client authentication credentials are exchanged in the second stage.

TTLS uses the TLS channel to exchange “attribute-value pairs” (AVPs), much like RADIUS. (In fact, the AVP encoding format is very similar to RADIUS.) The general encoding of information allows a TTLS server to validate AVPs against any type of authentication mechanism. TTLS implementations today support all methods defined by EAP, as well as several older methods (CHAP, PAP, MS-CHAP, and MS-CHAPv2).

TTLS can easily be extended to work with new protocols, by defining new attributes to support new protocols.

PEAP uses the TLS channel to protect a second EAP exchange. Authentication must be performed using a protocol that is defined for use with EAP. In practice, the restriction to EAP methods is not a severe drawback, because any important authentication protocol would be defined for use with EAP in short order, so that PEAP could use it. A far greater concern is client software support. PEAP is backed by Microsoft, and clients are available for recent professional versions of Windows. Suppliers of PEAP clients for other operating systems have yet to materialize, which may restrict PEAP usage to only pure Microsoft networks.

One major difference between TTLS and PEAP is that TTLS is much more widely implemented. TTLS products are available from multiple vendors and have been proven interoperable by a number of public demonstrations. TTLS software is also available for a wide range of client operating systems. **Table 4-3** provides a high-level functional comparison of the three common EAP types.

Table 4-3 EAP Type Functional Comparison

Extended Authentication Protocols	PROs	CONs
TLS	<ul style="list-style-type: none"> • IETF Standard 	<ul style="list-style-type: none"> • Certificate exchange happens in the clear • High maintenance overhead
TTLS	<ul style="list-style-type: none"> • Secure certificate exchange • Multivendor interoperability 	<ul style="list-style-type: none"> • Draft Standard only • Requires the user to have a PKI
PEAP	<ul style="list-style-type: none"> • Secure certificate exchange 	<ul style="list-style-type: none"> • U.S. networks only • Draft standard only

Selection of an authentication method is the key decision in securing networks. The authentication method drives the choice of authentication server, which in turn drives the choice of client software. Selecting an authentication method is a reasonably straightforward endeavor. Though there is not a large technical difference between the TTLS and PEAP protocols, TTLS has a number of slight advantages. In addition to a minor degree of flexibility at the protocol level, products are available now and support a much wider variety of client operating systems. **Table 4-4** provides a detailed comparison of the three protocols described in this section.

Table 4-4 Detailed Comparison of EAP Methods [13]

	TLS (RFC 2716)	TTLS (Internet Draft)	PEAP (Internet Draft)
Software			
Client implementations	Cisco, Funk, Meetinghouse, Microsoft, Open1x (open source)	Funk, Meetinghouse	Microsoft
Supported client platforms	Linux, Mac OS X, Windows 95/98/ME, Windows NT/2000/XP	Linux, Mac OS X, Windows 95/98/ME, Windows NT/2000/XP	Windows XP
Authentication server implementations by	Cisco, Funk, HP, FreeRADIUS (open source), Meetinghouse, Microsoft	Funk, Meetinghouse	Cisco
Authentication methods	Client certificates	Any	Any EAP method
Protocol Operations			
Basic protocol structure	Establish TLS session and validate certificates on both client and server	Two phases: (1) Establish TLS between client and TTLS server (2) Exchange attribute-value pairs between client and server	Two parts: (1) Establish TLS between client and PEAP server (2) Run EAP exchange over TLS tunnel
Fast session reconnect	No	Yes	Yes
WEP Integration	Server can supply WEP key with external protocol (e.g. RADIUS extension)		
PKI and Certificate Processing			
Server Certificate	Required	Required	Required
Client Certificate	Required	Optional	Optional
Cert Verification	Through certificate chain or Online Certificate Status Protocol (OCSP) TLS extension (current Internet draft)		
Effect of private key compromise	Reissue all server and client certificates	Re-issue certificates for servers (and clients, if using client certificates in first TLS exchange)	
Client and User Authentication			
Authentication direction	Mutual: Uses digital certificates both ways	Mutual: Certificate for server authentication, and tunneled method for client	Mutual: Certificate for server, and protected EAP method for client
Protection of user identity exchange	No	Yes; protected by TLS	Yes; protected by TLS

5 Security Applications of IEEE 802.1X

IEEE 802.1X can be used to secure existing networks and possibly add new services. Specifically, this discussion will focus on how 802.1X can support Ethernet, Token Ring, Fiber Distributed Data Interface (FDDI), 802.11 Networks, and virtual private networks (VPNs).

5.1 Ethernet and Token Ring/FDDI Networks

When 802.1X was drafted, the most popular wired networks being implemented were Ethernet, Token Ring, and FDDI. The committee decided to accommodate all of these networks. The standard specifies EAPOL frame formats for both Ethernet and Token Ring/FDDI networks. Token Ring and FDDI networks use the same token passing mechanism; however, Token Ring has a maximum speed of 16 Mbps, whereas an FDDI's is 100 Mbps. Defining frame formats for all the popular networks allows organizations to begin implementing the security features of the 802.1X, to immediately protect their networks. Furthermore, one of the goals of 802.1X is to allow for growth in the development of future networking technologies. The EAPOL-Key reserved a range of values that can be used to define newly developed keys on an as-needed basis.

5.2 802.11 Wireless Networks

Wireless networks continue to increase in popularity and implementation. Multiple facilities are providing public access points often referred to as "hot spots."⁷ Numerous organizations are considering this technology due to the increased speed that was offered by the 1999 revision of the IEEE 802.11 standard. Currently, there are three basic types of 802.11 networks, 802.11a, 802.11b, and 802.11g.

802.11b networks are the most popular, because they are the oldest and use the same technology as stipulated in the original standard. The difference today is that the maximum speed of an 802.11b network has increased from 2 Mbps to 11 Mbps. This puts them in line with many Ethernet networks, which have been established for years.

802.11a and 802.11g networks offer speed up to 54 Mbps and higher with recent software developments. 802.11g networks are backward compatible with 802.11b networks. They both operate in the 2.4 GHz RF range. As a result, an organization can gradually purchase new hardware, rather than committing to a large capital expenditure in order to take advantage of the new technology.

802.11a networks operate at the 5 GHz range of the RF spectrum. In order to use 802.11a technology, it is necessary to purchase new equipment. This new equipment needs to be either 802.11a or 802.11 a/b/g compatible. There are products on the market in one box that can be purchased to take advantage of the three technologies. The main advantage to an 802.11a network is that the 5 GHz range of the RF spectrum is not as crowded as the 2.4 GHz range. The 2.4 GHz range includes cordless phones, Bluetooth devices, 802.11b

⁷ A hot spot is a small geographic area of several hundred square feet in which you get access to an 802.11b wireless local area network. Hot Spots exists in homes, airport lounges, libraries, coffee shops, boardrooms, businesses, etc.

networks, and other products in development, such as Ultra Wide Band networking technologies.

Security has always been an issue with wireless networks. How can transmission media be secured when it is air? The original 802.11 standard did not address the issue of security. The standard was updated in 1999. That version did address security and created the Wired Equivalent Protocol (WEP). However, it was discovered that this protocol could be broken in a matter of hours. On May 30, 2001, the IEEE approved a Project Authorization Request (PAR) named 802.11i. The scope of the PAR was as follows:

“To enhance the current 802.11 MAC to provide improvements in security.”

The 802.11i Task Group (TG), also referred to as TGi, decided to take advantage of the recently approved 802.1X standard. The 802.1X standard was developed so that other EAPOL-Key fields could be created. The members of the 802.11i committee recognized that EAP was a good choice, because an organization could use any number of protocols, which provided implementers with options. Many standards dictate what must be used. As a result, a small firm may have to spend more money than needed in order to be compliant. TGi provided options with a requirement: whichever EAP is chosen, it should support mutual authentication. While the results of TGi have not been officially incorporated into the 802.11 standard as of this time, industry has already begun to implement the interim security fix known as Wi-Fi Protected Access (WPA). Many companies are producing equipment that is being sent to the Wi-Fi Alliance for testing and certification. Currently, the Wi-Fi Alliance has certified over 1,250 products as compatible.

5.3 Virtual Private Networks

A VPN is a secure, private communication tunnel between two or more devices across a public network (like the Internet). These VPN devices can be either a computer running VPN software or a special device like a VPN enabled router. The typical corporation today installs and maintains private lines leased from carriers, in order to link branch offices. Remote access servers provide service to telecommuters and mobile users, and users calling in from remote locations can incur long-distance charges. **Figure 5-1** illustrates a VPN between the office and home computer.

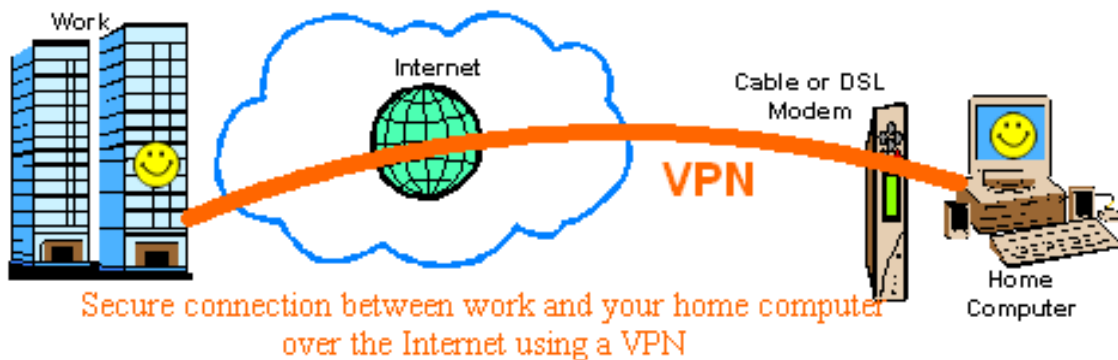


Figure 5-1 Example of VPN Connectivity [17]

Occasionally, a corporation runs leased lines to its customers, suppliers, and other business partners. Although such a link could provide better communications, the cost of the leased lines is often a barrier. However, many companies are able to justify the cost of an Internet connection through the need for e-mail, World Wide Web access, and other Internet services.

VPNs provide an alternative infrastructure that enables organizations to use the public Internet in a private, dedicated environment. With this approach, various corporate locations are connected via the Internet rather than over leased lines. The company's traffic is aggregated with other Internet traffic, so the whole system benefits from scale. Instead of calling into a private line connected to an access server, remote users can dial into an Internet service provider to communicate with others in the corporate local area network.

Even though a VPNs data travels across a public network like the Internet, it is secure because the data is encrypted via a very robust encryption algorithm. As a result, eavesdroppers can not understand the data unless they can decipher it.

The 802.1X standard allows for a number of EAPs which can be deployed on a VPN. Therefore, an organization can deploy a VPN without the burden and cost of doing a PKI rollout (both clients and servers would need authentication and certificates). There is no need for all the equipment in an organization (without PKI) to have digital certificates. Only the servers need them. The client, when it is authenticated, is issued a certificate by the server that is good for the session.

6 Observations and Summary

This TIB has addressed the development and content of IEEE Standard 802.1X for Local and Metropolitan Area Networks for Port-Based Network Access Control; discussed Extensible Authentication Protocols; and described and compared the three most common non-proprietary EAPs, TLS, TTLS, and PEAP. On that basis, the following observations and conclusions are presented:

- For the developers of the 802.1X standard, a principal objective was to develop a standard that worked with all current networking architectures e.g., Ethernet, Token Ring, and FDDI. They also provided the flexibility for future network architectures to be included in the standard.
- TLS, TTLS, and PEAP are all based on Secure Socket Layer developed by Netscape.
- 802.1X standard does not specify any particular EAP. However, it strongly recommends an EAP which supports mutual authentication.
- With over 60 EAPs currently available on the market, 802.1X can provide security at all levels, from personal to enterprise networks.
- The three most commonly implemented EAPs in corporate/government environments are TLS, TTLS, and PEAP.
- One of the most significant limitations of wireless networks in the past was securing transmission through the air. 802.11i Task Group has chosen 802.1X authentication to enhance the security of data transmitted by 802.11 wireless networks. This is part of an effort to update the 1999 802.11 standard to include WEP, which was found to be inadequate, because it was relatively easy to break. This task was completed in June 2004 [16].
- TLS is an EAP that is described in an IETF Request for Comment. TTLS and PEAP are currently IETF Drafts.
- TLS has the most strict security requirement, in that both the client and server must have valid certificates in order for authentication to occur.
- TTLS and PEAP provide security comparable to TLS, but support server-side certificates only. Therefore, some a priori information is necessary.
- TLS requires a full PKI rollout. TTLS and PEAP do not.
- Since the 802.1X standard can provide security comparable to a PKI rollout, many organizations may be able to take advantage of VPNs to connect their facilities, rather than going to the expense of leasing dedicated lines.
- While it cannot be used to protect classified activity, 802.1X can provide a mechanism for protecting sensitive information in support of NS/EP activities.

7 Recommendations

The issue of over-the-air security is of paramount importance to both the developers and users of wireless networks. This TIB has addressed this issue together with Extensible Authentication Protocols. In support of the NCS mission, as it relates to NS/EP and CIP, it is recommended that the NCS should:

- Conduct an evaluation of the various EAPs available and determine if any of them support its NS/EP and CIP missions.
- Provide requirements, in concert with the IETF PPP Extension Group, so that a potential EAP could be developed to facilitate the goals of the NCS mission, if research shows that none are currently available.
- Utilize an EAP which supports mutual authentication, such as TLS, TTLS, PEAP, or others that may surface.
- Consider using 802.1X to facilitate deployment of VPNs to allow for mobile or off-site communications needs, especially remote disaster recovery offices.
- As the NIST is continuing to evaluate and analyze EAPs and have not made any approvals or recommendations for usage, the NCS should monitor NIST activities to identify any EAPs which might support NCS requirements.
- Consider 802.1X as an authentication mechanism to support the deployment of wireless networks where appropriate.
- Track activities of IEEE, IETF to develop/approve further wireless standards, security standards, etc.

Appendix A: Acronyms

AAA	Authentication, Authorization, and Accounting
AES	Advanced Encryption Standard
AP	Access Point
AVP	Attribute Value Pairs
AVPS	Attribute Value Pairs
CCITT	International Telegraph and Telephone Consultative Committee
CHAP	Challenge-Handshake Authentication Protocol
CIP	Critical Infrastructure Protection
CSMA/CD	Carrier Sense Multiple Access - Collision Detection
DCE	Data Circuit-Terminating Equipment
DHS	Department of Homeland Security
DTE	Data Terminal Equipment
DTE/DCE	Data Terminal Equipment/Data Communications Equipment
E.O	Executive Order
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
EIA	Electronic Industries Association
EP	Electrophotographic Engine
FCS	Frame Check Sequence
FDDI	Fiber Distributed Data Interface
GHz	Gigahertz
HDLC	High-Level Data Link Control
HMAC	Hashed Message Authentications Code

HP	Hewlett-Packard
HTTP	Hyper-Text Transfer Protocol
HTTPS	Hyper-Text Transfer Protocol Secure
IAIP	Information Assurance and Infrastructure Protection
IANA	Internet Assigned Numbers Authority
ID	delay impairment value
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunication Union
IV	Initialization Vector
LAN	Local Area Networks
LAN/MAN	Local Area Network/Metropolitan Area Network
LCP	Link Control Protocol
LDAP	Lightweight Directory Access Protocol
LLC	Logical Link Control
MAC	Media Access Control
MD5	Message Digest 5
MS	Memory System
MU	Mauritius
NAK	Negative AcKnowledgegment
NCP	Network Control Protocols
NCS	National Communications System
NIST	National Institute of Standards and Technology
NS	National Security
NS/EP	National Security and Emergency Preparedness

NT	Network Termination
NTP	Network Termination Point
OCSP	Online Certificate Status Protocol
OS	Operating System
PAE	Port Access Entity
PAP	Packet Level Procedure
PAR	Project Authorization Request
PC	Personal Computer
PDU	Protocol Data Units
PEAP	Protected Extensible Authentication Protocol
PHY	Physical Layer Working Group of ATM Forum
PIN	Personal Identification Number
PKI	Public Key Infrastructure
PPP	Point to Point Protocol
PSN	Public Switched Network
RADIUS	Remote Access Dial-in User Server
RF	Radio Frequency
RFC	Request for Comments
RSA	Rivest, Shamir and Adleman Public Key Cryptosystem
SDE	Secure Data Exchange
SNMP	Simple Network Management Protocol
SSL	Secure Sockets Layer
STD	Set Direction Flag
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TG	Task Group
TIA	Telecommunications Industry Association
TIB	Technical Information Bulletin

TLS	Transport Layer Security
TR	Technical Report
TTLS	Tunneled Transport Layer Security
VPN	Virtual Private Networks
WEP	Wired Equivalent Protocol
WPA	Wi-Fi Protected Access

Appendix B: References

- [1] IEEE Standard 802.1X-2001, for Local and Metropolitan Area Networks Port-Based Network Access Control.
- [2] RFC 1305, Network Time Protocol (Version 3), Specification, Implementation and Analysis, March 1992.
- [3] RFC 2104, HMAC: Keyed-Hashing for Message Authentication, February 1997.
- [4] RFC 2716, PPP EAP TLS Authentication, October 1999.
- [5] RFC 2284, PPP Extensible Authentication Protocol (EAP), March 1998.
- [6] RFC 2869, RADIUS Extensions, June 2000.
- [7] RFC 1661, “The Point-to-Point Protocol,” July 1994.
- [8] IEEE 802.11i, TGi, “Draft Supplement to Standard for Telecommunications and Information Exchange between Systems – LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: “Specification for Enhanced Security,” Version 4.0, May 2003.
- [9] Internetworking Technologies Handbook, Cisco Press, 4th Edition, August 15, 2003.
- [10] Edney, Jon and Arbaugh, William A., “Real 802.11 Security: Wi-Fi Protected Access and 802.11i,” July 15, 2003.
- [11] Greenfield, David, “SSL and TLS,” Network Magazine, December 4, 2002.
- [12] PEAP, 12/2003, Symbol, <http://www.symbol.com/products/wireless/peap.html>
- [13] Gast, Matthew, “A Technical Comparison of TTLS and PEAP,” O’Reilly Media, October 2002.
- [14] “Token Ring,” Network Cabling Help, www.datacottage.com/nch/troperation.htm
- [15] “Token Ring/FDDI Networks,” RAD Data Communications Ltd, www2.rad.com/networks/1997/nettut/token_ring.html
- [16] IEEE STD 802.11i-2004, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements, June 24, 2004.
- [17] “VPN – Virtual Private Networking, an Overview.” www.homenethelp.com/vpn
- [18] RFC 2759, “PPP Challenge Handshake Authentication Protocol (CHAP),” August 1966.
- [19] RFC 3748, “Extensible Authentication Protocol,” June 2004.
- [20] Internet Draft, “Tunneled TLS Authentication Protocol Version 1.0,” February 2005.
- [21] RFC 2246, “TLS Protocol Version 1.0,” January 1999. This RFC has been updated by RFC 3546, “TLS Extension,” June 2003.

Appendix C: Bibliography

IEEE STD 802: Overview and Architecture.

ANSI/IEEE STD 802.1B and 802.1K [ISO/IEC 15802-2]: LAN/MAN Management.

ANSI/IEEE STD 802.1D: Media Access Control (MAC) Bridges.

ANSI/IEEE STD 802.1E [ISO/IEC 15802-5]: System Load Protocol.

ANSI/IEEE STD 802.1F: Common Definitions and Procedures for IEEE 802 Management Information.

ANSI/IEEE STD 802.1G [ISO/IEC 11802-5]: Remote Media Access Control (MAC) Bridging.

ANSI/IEEE STD 802.1H [ISO/IEC TR 11802-5]: Recommended Practice for Media Access Control (MAC).

ANSI/IEEE STD 802.1Q [ISO/IEC TR 11802-5: Virtual Bridged Local Area Networks.

ANSI/IEEE STD 802.2 [ISO/IEC 8802-2]: Logical Link Control.

ANSI/IEEE STD 802.3 [ISO/IEC 8802-3]: CSMA/CD Access Method and Physical Layer Specifications.

ANSI/IEEE STD 802.4 [ISO/IEC 8802-4]: Token Bus Access Method and Physical Layer Specifications.

ANSI/IEEE STD 802.5 [ISO/IEC 8802-5]: Token Ring Access Method and Physical Layer Specifications.

ANSI/IEEE STD 802.6 [ISO/IEC 8802-6]: Distributed Queue Dual Bus Access Method and Physical Layer Specifications.

ANSI/IEEE STD 802.10: Interoperable LAN/MAN Security. Currently Approved: Secure Data Exchange (SDE).

ANSI/IEEE STD 802.11 [ISO/IEC 8802-11]: Wireless LAN Medium Access Control (MAC) Sublayer and Physical Layer Specifications.

ANSI/IEEE STD 802.12 [ISO/IEC 8802-12]: Demand Priority Access Method, Physical Layer and Repeater Specification.

IEEE STD 802.15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks.

IEEE STD 802.16: Standard Air Interface for Fixed Broadband Wireless Access Systems.

IEEE STD 802.17: Resilient Packet Ring Access Method and Physical Layer Specifications.

IEEE STD 802.7: IEEE Recommended Practice for Broadband Local Area Networks.

RADIUS By Jonathan Hassell Publisher: O'Reilly Pub Date: October 2002.

802.11 Wireless Networks: The Definitive Guide By Matthew Gast, O'Reily Publisher, April 2002.

802.11 Security By Bob Fleck, Bruce Potter, O'Reilly Publisher, December 2002.

Wireless Hacks By Rob Flickenger, O'Reilly Publisher, September 2003.

Maximum Security, Third Edition by Anonymous Publisher: Sams Publishing, April 1, 2001.

IETF Draft, "EAP Tunneled Authentication Protocol Version 1 (TTLSv1)," February 2005.

IETF Draft, "Protected EAP Protocol (PEAP) Version 2," October 2004.

"Apache HTTP Server Version 2.0," Apache Software Foundation, 1999-2004.