Chapter 10
Security Protocols of the Data Link Layer

- IEEE 802.1x
- Point-to-Point Protocol (PPP)
- Point-to-Point Tunneling Protocol (PPTP)
Scope of Link Layer Security Protocols

- According to the classical understanding of the OSI model, the link layer provides an assured data transmission service between two peer entities that are directly inter-connected by a communications medium.

- Its main tasks are:
  - Error detection and correction
  - Medium access control (MAC, not to be mixed up with message authentication code) for shared media, e.g. Ethernet, etc.

- Not all of today’s networking technology fits nicely into that model:
  - Dial-up connections to an Internet service provider
  - Virtual Private Network (VPN) solutions

- In this class, we content ourselves with the following definition:
  - The purpose of a link layer security protocol is to ensure specific security properties of link layer PDUs, that is the PDUs of the protocol layer carrying the PDUs of the network layer (e.g. IP)
IEEE 802.1x: Background & Goals

- The Institute of Electrical and Electronics Engineers (IEEE) 802 LAN/MAN Standards Committee develops local area network standards and metropolitan area network standards.

- The most widely used standards are:
  - Ethernet family (802.3, generally referred to as CSMA/CD),
  - Token Ring (802.5),
  - Wireless LAN (802.11)

- The IEEE committee is currently working on a standard that:
  - aims to “restrict access to the services offered by a LAN to those users and devices that are permitted to make use of those services”
  - may be used with different IEEE 802.x technologies
  - defines port based network access control to provide a means of “authenticating and authorizing devices attached to a LAN port that has point-to-point connection characteristics”
  - is generally referred to as IEEE 802.1x
IEEE 802.1x introduces the notion of two logical ports:
- the uncontrolled port allows to authenticate a device
- the controlled port allows an authenticated device to access LAN services
IEEE 802.1x: Roles

- Three principal roles are distinguished:
  - A device that wants to use the service offered by an IEEE 802.1x LAN acts as a *supplicant* requesting access to the controlled port.
  - The point of attachment to the LAN infrastructure (e.g. a MAC bridge) acts as the * authenticator* demanding the supplicant to authenticate itself.
  - The authenticator does not check the credentials presented by the supplicant itself, but passes them to his *authentication server* for verification.

- Accessing a LAN with IEEE 802.1x security measures:
  - Prior to successful authentication the supplicant can access the uncontrolled port:
    - The port is uncontrolled in the sense, that it allows access prior to authentication.
    - However, this port allows only restricted access.
  - Authentication can be initiated by the supplicant or the authenticator.
  - After successful authentication the controlled port is opened.
IEEE 802.1x Security Protocols & Message Exchange

- IEEE 802.1x does not define its own security protocols, but advocates the use of existing protocols:
  - The Extensible Authentication Protocol (EAP) may realize basic device authentication [RFC 2284]
  - If negotiation of a session key during authentication is required, the use of the PPP EAP TLS Authentication Protocol is recommended [RFC 2716]
  - Furthermore, the authentication server is recommended to be realized with the Remote Authentication Dial In User Service (RADIUS) [RFC 2865]
- Exchange of EAP messages between supplicant and authenticator is realized with the EAP over LANs (EAPOL) protocol:
  - EAPOL defines the encapsulation techniques that shall be used in order to carry EAP packets between supplicant port access entities (PAE) and Authenticator PAEs in a LAN environment
  - EAPOL frame formats have been defined for various members of the 802.x protocol family, e.g. EAPOL for Ethernet, ...
  - Between supplicant and authenticator RADIUS messages may be used
IEEE 802.1x: Example of an 802.1x Authentication

Supplicant PAE  Authenticator PAE  Authentication Server

EAPOL-Start

EAP-Request/Identity

EAP-Response/Identity(MyID)

EAP-Request/OTP
OTP Challenge

EAP-Request/OTP
OTP Passwd

EAP-Success
Port authorized
Authentication successfully completed

[source: IEEE Draft P802.1X/D11]
Point-to-Point Protocol: Purpose and Tasks

- Large parts of the Internet rely on point-to-point connections:
  - Wide area network (WAN) connections between routers
  - Dial-up connections of hosts using modems and telephone lines

- Protocols for this purpose:
  - Serial Line IP (SLIP): no error detection, supports only IP, no dynamic address assignment, no authentication [RFC 1055]
  - Point-to-Point Protocol (PPP): successor to SLIP, supports IP, IPX, ...

- PPP [RFC 1661/1662]:
  - Layer-2 frame format with frame delimitation and error detection
  - Control protocol \((\text{Link Control Protocol}, LCP)\) for connection establishment, -test, -negotiation, and -release
  - Separate \(\text{Network Control Protocols (NCP)}\) for supported Layer-3 protocols
## Point-to-Point Protocol: Packet Format

<table>
<thead>
<tr>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Protocol</th>
<th>Payload</th>
<th>Checksum</th>
<th>Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td>01111110</td>
<td>11111111</td>
<td>00000011</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Character-oriented (instead of bit-oriented) \(\Rightarrow\) byte aligned frames
- Code transparency achieved through character stuffing
- Usually only unnumbered frames are transmitted, however, in scenarios with high error probability (wireless communications) a more reliable mode with sequence numbers and re-transmissions can be negotiated
- Supported protocols for the payload field are, among others: IP, IPX, Appletalk
- If not otherwise negotiated the maximum payload size is 1500 byte
- Additional negotiation supports smaller packet headers
Point-to-Point Protocol: A Typical PPP Connection

- Usage Scenario “Internet access of a PC via modem”:
  - User calls *Internet service provider (ISP)* via modem and establishes a “physical” connection via the plain old telephone service (POTS)
  - Caller sends multiple LCP-packets in PPP-frames to choose desired PPP-parameters
  - Security specific negotiation (see below)
  - Exchange of NCP-packets to configure network layer:
    - e.g. configuration of IP including dynamic allocation of an IP address via Dynamic Host Configuration Protocol (DHCP)
  - Caller may use arbitrary Internet services like any other host with a fixed connection to the Internet
  - For connection termination the allocated IP address and the network layer connection are released
  - The layer-2 connection is released via LCP and the modem closes down the “physical” connection
Point-to-Point Protocol: Link Control Protocol

- Frame format of the *Link Control Protocol (LCP)*:
  - **Code**: configure-request, configure-ack, configure-nack, configure-reject, terminate-request, terminate-ack, code-reject, protocol-reject, echo-request, echo-reply, discard-request
  - **Length**: indicates the length of the LCP-packet including the code field etc.
  - **Data**: zero or more octets of command-specific data

![Frame format of LCP](image)

- The *configure* primitives of LCP allow to configure the link layer:
  - There exist various options for this primitive for configuration of different aspects (max. receive unit, protocol compression, authentication, ...)

[NetSec], WS 2005/06 10.11
Point-to-Point Protocol: Security Services

- The original version of PPP [RFC 1661] suggests the optional run of an authentication protocol after the link establishment phase:
  - If required, authentication is demanded by one peer entity via an LCP Configuration-Request at the end of the link establishment phase
  - Originally, two authentication protocols have been defined:
    - Password Authentication Protocol (PAP)
    - Challenge Handshake Authentication Protocol (CHAP)
  - Meanwhile, an extensible protocol has been defined:
    - Extensible Authentication Protocol (EAP)
    - PPP EAP Transport Level Security Protocol (PPP-EAP-TLS)
- Furthermore, encryption can be negotiated after authentication:
  - Protocols:
    - Encryption Control Protocol (ECP) for negotiation
    - PPP DES Encryption Protocol (DESE)
    - PPP Triple DES Encryption Protocol (3DESE)
Point-to-Point Protocol: Authentication Protocols (1)

- Password Authentication Protocol (PAP):
  - PAP was defined 1992 in RFC 1334
  - The protocol is very simple:
    - Prerequisite: the authenticator knows a password of the peer entity
    - At the end of the link establishment phase one entity, called authenticator, demands the peer entity to authenticate with PAP
    - The peer entity sends an authenticate-request message containing its’ peer ID and password
    - The authenticator checks if the provided information is correct and answers with either an authenticate-ack or an authenticate-nack
  - As the protocol provides no cryptographic protection, it is insecure
  - PAP is not mentioned in updated RFCs for PPP authentication [RFC1994]
Point-to-Point Protocol: Authentication Protocols (2)

- **Challenge Handshake Authentication Protocol (CHAP):**
  - CHAP is also defined in RFC 1334 and RFC 1994
  - It realizes a simple challenge-response protocol:
    - Prerequisite: authenticator and peer entity share a secret
    - After the link establishment phase the authenticator (A) sends a challenge message containing an *identifier* for this challenge, a random number $r_A$, and its name to the peer entity (B):
      \[ A \rightarrow B: (1, \text{identifier}, r_A, A) \]
    - The peer entity computes a cryptographic hash function over its name, the shared secret $K_{A,B}$ and the challenge random number $r_A$ and sends the following message:
      \[ B \rightarrow A: (2, \text{identifier}, H(B, K_{A,B}, r_A), B) \]
    - Upon reception of this message the authenticator re-computes the hash value and compares it with the received one; if both values match it answers with a *success* message
    - RFC 1994 specifies, that MD5 must be supported as hash function, but use of other hash functions can be negotiated
Extensible Authentication Protocol (EAP):

- EAP is a general protocol for PPP authentication which supports multiple authentication methods [RFC2284]
- The main idea behind EAP is to provide a common protocol to run more elaborate authentication methods than “1 question + 1 answer”
- The protocol provides basic primitives:
  - Request, Response: further refined by type field + type specific data
  - Success, Failure: to indicate the result of an authentication exchange
- Type fields:
  - Identity
  - Notify
  - Nak (response only, to answer unacceptable request types)
  - MD5 Challenge (this corresponds to CHAP)
  - One-Time Password (OTP): defined in [RFC2289]
  - Generic Token Card
  - EAP-TLS
One-Time Password (OTP):

- The basic idea of OTP is to transmit a “password”, that can only be used for one run of an authentication dialogue.

Initial Setup:

- The authenticator $A$ sends a seed value $r_A$ and the peer entity $B$ concatenates it with his password and computes a hash value: $PW_N = H^N(r_A, \text{password}_B)$
- The pair $(N, PW_N)$ is “securely” transmitted to the authenticator and stored at the authenticator.

Authentication dialogue:

- $A \rightarrow B$: $N - 1$
- $B \rightarrow A$: $PW_{N-1} := H^{N-1}(r_A, \text{password}_B)$
- A checks if $H(PW_{N-1}) = PW_N$, and stores $(N - 1, PW_{N-1})$ as the new authentication information for B.

Security: In order to break this scheme, an attacker would have to eavesdrop one $PW_N$ and compute $H^{-1}(PW_N)$ which is impractical.
Generic Token Card:
- Basically, a challenge response dialogue
- A token card is used to compute a response to a challenge:
  - The challenge is presented to the user who has to type it to his token card device
  - The token card computes and displays the response
  - The user enters the response into the system that sends it as an answer to the challenge message

PPP-EAP-TLS:
- TLS stands for *Transport Layer Security* [RFC 2246]
- Thus, the authentication dialogue of TLS is run
- This dialogue will be explained in detail in chapter 12 on transport layer security
After the link establishment and the authentication phase, encryption can be negotiated for a PPP connection:

- The *Encryption Control Protocol (ECP)* [RFC1968] is responsible for configuring and enabling data encryption algorithms on both ends of the PPP link:
  - ECP uses the same frame format as LCP and introduces two new primitives: Reset-Request and Reset-Ack for indicating decryption errors independently for each direction (useful for cryptographic resynchronization)
  - A specific encryption method is negotiated using the *configure* primitive containing an option specifying *DESE, 3DESE, Proprietary*, etc.
  - Proprietary encryption protocols are identified by a registered *organizational unit identifier (OUI)* + a vendor specific value
  - Exactly one ECP packet is transported in the PPP information field of a link layer packet
  - ECP packets are identified by the PPP protocol field:
    - 0x8053 for “standard” operation
    - 0x8055 for individual link data encryption on multiple links to the same destination
Point-to-Point Protocol: Encryption Protocols (2)

- **PPP DESE v2 (DESEv2) [RFC2419]**
  - *Sequence Number*: initially 0, this number is incremented by the encrypting entity with every packet sent
  - Ciphertext: the encrypted protocol and information fields of a PPP packet
    - messages are padded to a multiple of 8 octets prior to encryption
    - encryption is realized with DES in CBC mode

- **PPP 3DES Encryption Protocol (3DESE):**
  - PPP 3DESE [RFC2420] is very similar to the PPP DESE
  - Encryption of PPP payload is like DESE, with the difference that 3DES is used with 3 different keys

- All of the PPP encryption protocols assume, that a session key for encryption / decryption of PPP packets has been agreed upon prior to the encryption phase:
  - This assumption is reasonable, as session key establishment is a task that should be fulfilled during the authentication phase
  - However, only the PPP-EAP-TLS authentication protocol supports session key establishment
Point to Point Tunneling Protocol (PPTP)

- PPP was originally designed to be run between “directly” connected entities, that is entities which share a layer-2 connection
  - Example: a PC and a dialup-router of an Internet service provider connected over the telephone network using modems

- The basic idea of PPTP is to extend the protocol’s reach over the entire Internet by defining transport of PPP PDUs in IP packets
  - Thus, the payload of PPTP PDUs are PPP packets (without layer-2 specific fields like HDLC flags, bit insertion, control characters, CRC error check values, etc.)

- PPP packets are encapsulated in GRE packets (generic routing encapsulation) that themselves are encapsulated in IP packets:

```
<table>
<thead>
<tr>
<th>Media Header (e.g. Ethernet MAC header)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Header</td>
</tr>
<tr>
<td>GRE V.2 Header</td>
</tr>
<tr>
<td>PPP Packet</td>
</tr>
</tbody>
</table>
```
PPTP: Voluntary vs. Compulsory Tunneling

- PPTP realizes a “tunnel” over the Internet that carries PPP packets
- Such a tunnel can be realized between different entities:
  - A client PC and a PPTP Remote Access Server (RAS):
    - This is also referred to as voluntary tunneling, as the client PC is actively participating in the PPTP processing
    - This variant allows to support secure communication between a client PC and a specific subnetwork using any access and intermediate network(s)
  - An ISP’s Point of Presence (POP) and a PPTP Remote Access Server:
    - This is also referred to as compulsory tunneling, as the client PC is not involved in the decision whether PPTP will be used or not
    - This allows to realize security on the subnetwork level but does not realize true end-to-end security between the client PC and the RAS
    - In compulsory tunneling the ISP POP acts as a proxy client to the RAS
PPTP: Voluntary Tunneling Protocol Layers

Client ISP POP PPTP-Tunnel PPTP RAS Application Server

IP / IPX / NetBEUI packet flow

PPP

IP / IPX / NetBEUI
PPP
GRE Version 2
IP
PPP
PPP Framing (HDLC)

Physical Layer

Layer 2 (e.g. 802.x)

Physical Layer

Physical Layer

Layer 2

Physical Layer

PPP Framing (HDLC)
PPTP: Compulsory Tunneling Protocol Layers

Client
ISP POP
PPTP-Tunnel
PPTP RAS
Application Server

IP / IPX / NetBEUI packet flow

PPP
PPTP

<table>
<thead>
<tr>
<th>IP / IPX / NetBEUI</th>
<th>IP / IPX / NetBEUI</th>
<th>IP / IPX / NetBEUI</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPP</td>
<td>PPP</td>
<td>Layer 2 (e.g. 802.x)</td>
</tr>
<tr>
<td>PPP Framing (e.g. HDLC)</td>
<td>GRE Version 2</td>
<td>Layer 2</td>
</tr>
<tr>
<td>Physical Layer</td>
<td>IP</td>
<td>Physical Layer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
PPTP: Voluntary Tunneling Packet Construction at Client

- Application → User Data
- TCP/IP Stack → IP TCP/UDP User Data
- PPTP Software → GRE PPP IP TCP/UDP User Data
- TCP/IP Stack → IP GRE PPP IP TCP/UDP User Data
- PPP Device Driver → PPP Framing PPP IP GRE PPP IP TCP/UDP User Data
PPTP / PPP Proprietary Extensions & Some “History”

- PPTP has been largely deployed as a consequence of Microsoft’s support for it:
  - It has been developed with Microsoft’s active involvement and is documented in [RFC2637]
  - Microsoft implemented it as a part of its Remote Access Service (RAS)
- Microsoft further specified “proprietary” extensions for PPP:
  - Microsoft PPP CHAP Extensions [RFC2433]
  - Microsoft Point to Point Encryption Protocol [RFC3078]
- However, a series of vulnerabilities have been discovered in PPTP version 1 and also in an improved version 2 [SM98a, SMW99a]:
  - A general consensus to adopt PPTP as a standard protocol could not be reached in the IETF working groups
  - Furthermore, a similar protocol (Layer 2 Forwarding, L2F) had been proposed by Cisco as a competing approach
  - As a consequence, a compromise was found to merge the advantages of both proposals into one single protocol Layer 2 Tunneling Protocol (L2TP)
Additional References (1)


Additional References (2)


