802.1X Port−Based Authentication HOWTO

Lars Strand

<lars strand (at) gnist org>

2004−08−18

Revision History

Revision 1.0	2004−10−18	Revised by: LKS
Initial Release, reviewed by TLDP.		
Revision 0.2b	2004−10−13	Revised by: LKS
Various updates. Thanks to Rick Moen <rick (at) linuxmafia com> for language review.		
Revision 0.0	2004−07−23	Revised by: LKS
Initial draft.		

This document describes the software and procedures to set up and use IEEE 802.1X Port−Based Network Access Control using Xsupplicant as Supplicant with FreeRADIUS as a back−end Authentication Server.

Table of Contents

Table of Contents

1. Introduction

This document describes the software and procedures to set up and use 802.1X: Port−Based Network Access Control using Xsupplicant with PEAP (PEAP/MS−CHAPv2) as authentication method and FreeRADIUS as back−end authentication server.

If another authentication mechanism than PEAP is preferred, e.g., EAP−TLS or EAP−TTLS, only a small number of configuration options needs to be changed. PEAP/MS−CHAPv2 are also supported by Windows XP SP1/Windows 2000 SP3.

1.1. What is 802.1X?

The 802.1X−2001 standard states:

"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 has point−to−point connection characteristics, and of *preventing access* to that port in cases which the authentication and authorization fails. A port in this context is a single point of attachment to the LAN infrastructure." −−− 802.1X−2001, page 1.

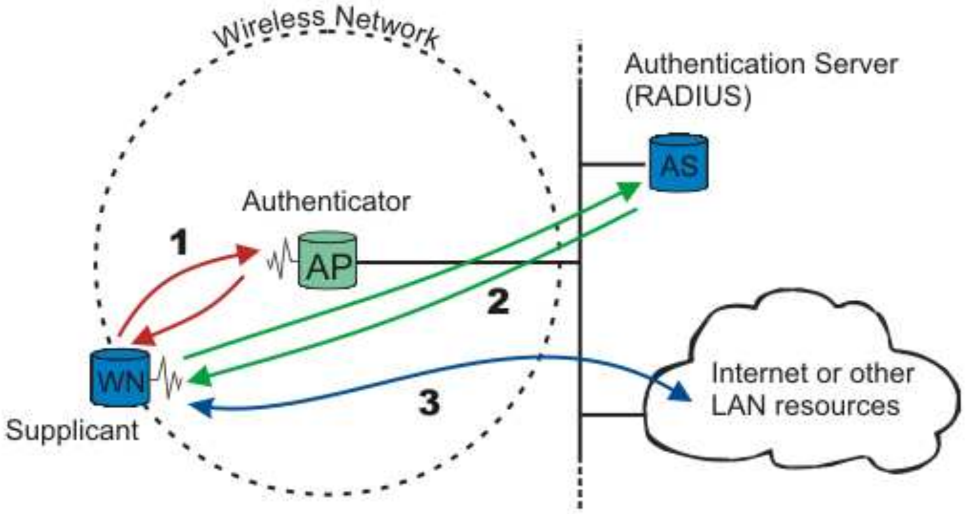

Figure 802.1X: A wireless node must be authenticated before it can gain access to other LAN resources.

1. When a new wireless node (WN) requests access to a LAN resource, the access point (AP) asks for the WN's identity. *No other traffic than EAP is allowed before the WN is authenticated (the "port" is closed).*

   The wireless node that requests authentication is often called *Supplicant*, although it is more correct to say that the wireless node *contains* a Supplicant. The Supplicant is responsible for responding to Authenticator data that will establish its credentials. The same goes for the access point; the *Authenticator is* not the access point. Rather, the access point contains an Authenticator. The Authenticator does not even need to be in the access point; it can be an external component.

EAP, which is the protocol used for authentication, was originally used for dial−up PPP. The identity was the username, and either PAP or CHAP authentication [RFC1994] was used to check the user's password. Since the identity is sent in clear (not encrypted), a malicious sniffer may learn the user's identity. "Identity hiding" is therefore used; the real identity is not sent before the encrypted TLS tunnel is up.

2. After the identity has been sent, the authentication process begins. The protocol used between the Supplicant and the Authenticator is EAP, or, more correctly, EAP encapsulation over LAN (EAPOL). The Authenticator re−encapsulates the EAP messages to RADIUS format, and passes them to the Authentication Server.

   During authentication, the Authenticator just relays packets between the Supplicant and the Authentication Server. When the authentication process finishes, the Authentication Server sends a success message (or failure, if the authentication failed). *The Authenticator then opens the "port" for the Supplicant.*
3. After a successful authentication, the Supplicant is granted access to other LAN resources/Internet.

See figure 802.1X for explanation.

Why is it called "port"−based authentication? The Authenticator deals with *controlled* and *uncontrolled* ports. Both the controlled and the uncontrolled port are logical entities (virtual ports), but use the same physical connection to the LAN (same point of attachment).

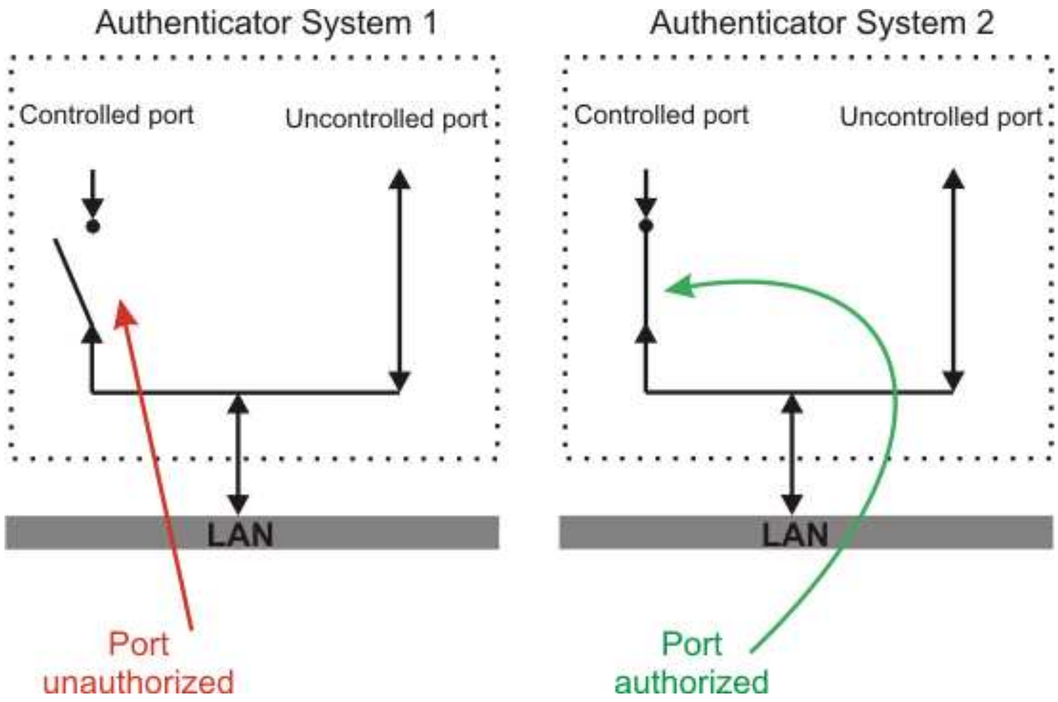

Figure port: The authorization state of the controlled port.

Before authentication, only the uncontrolled port is "open". The only traffic allowed is EAPOL; see Authenticator System 1 on figure port. After the Supplicant has been authenticated, the controlled port is opened, and access to other LAN resources are granted; see Authenticator System 2 on figure port.

802.1X plays a major role in the new IEEE wireless standard 802.11i.

1.2. What is 802.11i?

1.2.1. WEP

Wired Equivalent Privacy (WEP), which is part of the original 802.11 standard, should provide confidentiality. Unfortunately WEP is poorly designed and easily cracked. There is no authentication mechanism, only a weak form of access control (must have the shared key to communicate). Read more here.

As a response to WEP broken security, IEEE has come up with a new wireless security standard named 802.11i. 802.1X plays a major role in this new standard.

1.2.2. 802.11i

The new security standard, 802.11i, which was ratified in June 2004, fixes all WEP weaknesses. It is divided into three main categories:

1. *Temporary Key Integrity Protocol (TKIP)* is a short−term solution that fixes all WEP weaknesses. TKIP can be used with old 802.11 equipment (after a driver/firmware upgrade) and provides integrity and confidentiality.
2. *Counter Mode with CBC−MAC Protocol (CCMP) [RFC2610]* is a new protocol, designed from ground up. It uses AES [FIPS 197] as its cryptographic algorithm, and, since this is more CPU intensive than RC4 (used in WEP and TKIP), new 802.11 hardware may be required. Some drivers can implement CCMP in software. CCMP provides integrity and confidentiality.
3. *802.1X Port−Based Network Access Control:* Either when using TKIP or CCMP, 802.1X is used for authentication.

In addition, an optional encryption method called "Wireless Robust Authentication Protocol" (WRAP) may be used instead of CCMP. WRAP was the original AES−based proposal for 802.11i, but was replaced by CCMP since it became plagued by property encumbrances. Support for WRAP is optional, but CCMP support is mandatory in 802.11i.

802.11i also has an extended key derivation/management, described next.

1.2.3. Key Management

1.2.3.1. Dynamic key exchange and management

To enforce a security policy using encryption and integrity algorithms, keys must be obtained. Fortunately, 802.11i implements a key derivation/management regime. See figure KM.

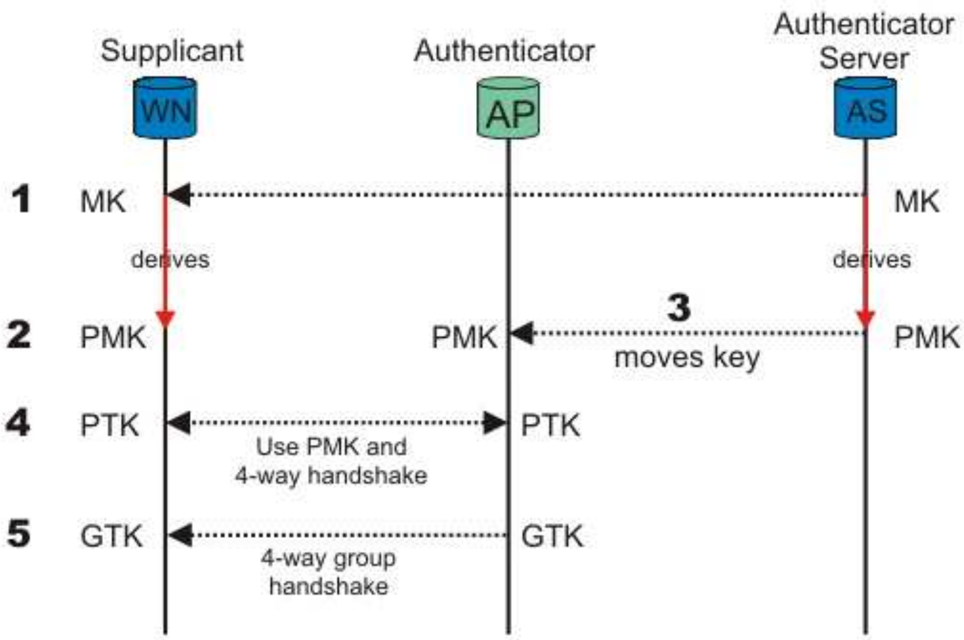

Figure KM: Key management and distribution in 802.11i.

1. When the Supplicant (WN) and Authentication Server (AS) authenticate, one of the last messages sent from AS, given that authentication was successful, is a *Master Key (MK)*. After it has been sent, the MK is known only to the WN and the AS. The MK is bound to this session between the WN and the AS.
2. Both the WN and the AS derive a new key, called the *Pairwise Master Key (PMK)*, from the Master Key.
3. The PMK is then moved from the AS to the Authenticator (AP). Only the WN and the AS can derive the PMK, else the AP could make access−control decisions instead of the AS. The PMK is a fresh symmetric key bound to this session between the WN and the AP.
4. PMK and a 4−way handshake are used between the WN and the AP to derive, bind, and verify a *Pairwise Transient Key (PTK)*. The PTK is a collection of operational keys:

    ♦ *Key Confirmation Key (KCK)*, as the name implies, is used to prove the posession of the PMK and to bind the PMK to the AP.
    ♦ *Key Encryption Key (KEK)* is used to distributed the Group Transient Key (GTK). Described below.
    ♦ *Temporal Key 1 & 2 (TK1/TK2)* are used for encryption. Usage of TK1 and TK2 is ciphersuite−specific.

    See figure PKH for a overview of the Pairwise Key Hierarchy.
5. The KEK and a 4−way group handshake are then used to send the *Group Transient Key (GTK)* from the AP to the WN. The GTK is a shared key among all Supplicants connected to the same Authenticator, and is used to secure multicast/broadcast traffic.

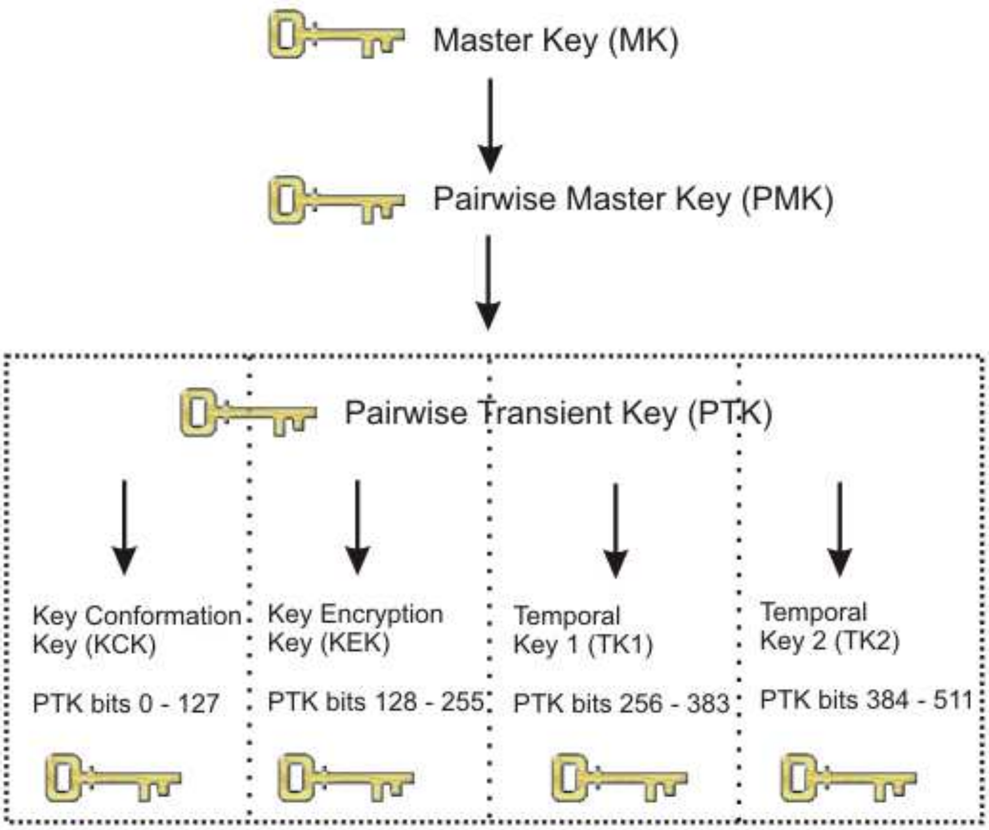

Figure PKH: Pairwise Key Hierarchy

1.2.3.2. Pre−shared Key

For small office / home office (SOHO), ad−hoc networks or home usage, a pre−shared key (PSK) may be used. When using PSK, the whole 802.1X authentication process is elided. This has also been called "WPA Personal" (WPA−PSK), whereas WPA using EAP (and RADIUS) is "WPA Enterprise" or just "WPA".

The 256−bit PSK is generated from a given password using PBKDFv2 from [RFC2898], and is used as the Master Key (MK) described in the key management regime above. It can be one single PSK for the whole network (insecure), or one PSK per Supplicant (more secure).

1.2.4. TSN (WPA) / RSN (WPA2)

The industry didn't have time to wait until the 802.11i standard was completed. They wanted the WEP issues fixed now! Wi−Fi Alliance felt the pressure, took a "snapshot" of the standard (based on draft 3), and called it *Wi−Fi Protected Access (WPA)*. One requirement was that existing 802.11 equipment could be used with WPA, so WPA is basically TKIP + 802.1X.

WPA is not the long term solution. To get a *Robust Secure Network (RSN)*, the hardware must support and use CCMP. RSN is basically CCMP + 802.1X.

RSN, which uses TKIP instead of CCMP, is also called Transition Security Network (TSN). RSN may also be called WPA2, so that the market don't get confused.

Confused?

Basically:

- TSN = TKIP + 802.1X = WPA(1)
- RSN = CCMP + 802.1X = WPA2

In addition comes key management, as described in the previous section.

1.3. What is EAP?

Extensible Authentication Protocol (EAP) [RFC 3748] is just the transport protocol optimized for authentication, not the authentication method itself:

" [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 IP. EAP provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees. Fragmentation is not supported within EAP itself; however, individual EAP methods may support this." −−− RFC 3748, page 3

1.4. EAP authentication methods

Since 802.1X is using EAP, multiple different authentication schemes may be added, including smart cards, Kerberos, public key, one time passwords, and others.

Some of the most−used EAP authentication mechanism are listed below. A full list of registered EAP authentication types is available at IANA: http://www.iana.org/assignments/eap−numbers.

Not all authentication mechanisms are considered secure!

- *EAP−MD5:* MD5−Challenge requires username/password, and is equivalent to the PPP CHAP protocol [RFC1994]. This method does not provide dictionary attack resistance, mutual authentication, or key derivation, and has therefore little use in a wireless authentication enviroment.
- *Lightweight EAP (LEAP):* A username/password combination is sent to a Authentication Server (RADIUS) for authentication. Leap is a proprietary protocol developed by Cisco, and is not considered secure. Cisco is phasing out LEAP in favor of PEAP. The closest thing to a published standard can be found here.
- *EAP−TLS:* Creates a TLS session within EAP, between the Supplicant and the Authentication Server. Both the server and the client(s) need a valid (x509) certificate, and therefore a PKI. This method provides authentication both ways. EAP−TLS is described in [RFC2716].
- *EAP−TTLS:* Sets up a encrypted TLS−tunnel for safe transport of authentication data. Within the TLS tunnel, (any) other authentication methods may be used. Developed by Funk Software and Meetinghouse, and is currently an IETF draft.
- *Protected EAP (PEAP):* Uses, as EAP−TTLS, an encrypted TLS−tunnel. Supplicant certificates for both EAP−TTLS and EAP−PEAP are optional, but server (AS) certificates are required. Developed by Microsoft, Cisco, and RSA Security, and is currently an IETF draft.

* *EAP−MSCHAPv2:* Requires username/password, and is basically an EAP encapsulation of MS−CHAP−v2 [RFC2759]. Usually used inside of a PEAP−encrypted tunnel. Developed by Microsoft, and is currently an IETF draft.

1.5. What is RADIUS?

Remote Authentication Dial−In User Service (RADIUS) is defined in [RFC2865] (with friends), and was primarily used by ISPs who authenticated username and password before the user got authorized to use the ISP's network.

802.1X does not specify what kind of back−end authentication server must be present, but RADIUS is the "de−facto" back−end authentication server used in 802.1X.

There are not many AAA protocols available, but both RADIUS and DIAMETER [RFC3588] (including their extensions) conform to full AAA support. AAA stands for Authentication, Authorization, and Accounting (IETF's AAA Working Group).

2. Obtaining Certificates

OpenSSL must be installed to use either EAP−TLS, EAP−TTLS, or PEAP!

When using EAP−TLS, both the Authentication Server and all the Supplicants (clients) need certificates [RFC2459] . Using EAP−TTLS or PEAP, only the Authentication Server requires certificates; Supplicant certificates are optional.

You get certificates from the local certificate authority (CA). If there is no local CA available, OpenSSL may be used to generate self−signed certificates.

Included with the FreeRADIUS source are some helper scripts to generate self−signed certificates. The scripts are located under the `scripts/` folder included with the FreeRADIUS source:

`CA.all` is a shell script that generates certificates based on some questions it ask. `CA.certs` generates certificates non−interactively based on pre−defined information at the start of the script.

The scripts uses a Perl script called `CA.pl`, included with OpenSSL. The path to this Perl script in `CA.all` and `CA.certs` may need to be changed to make it work.

More information on how to generate your own certificates can be found in the SSL certificates HOWTO.

3. Authentication Server: Setting up FreeRADIUS

FreeRADIUS is a fully GPLed RADIUS server implementation. It supports a wide range of authentication mechanisms, but PEAP is used for the example in this document.

3.1. Installing FreeRADIUS

Installing FreeRADIUS

1. Head over to the FreeRADIUS site, http://www.freeradius.org/, and download the latest release.

   ```
   # cd /usr/local/src
      # wget ftp://ftp.freeradius.org/pub/radius/freeradius-1.0.0.tar.gz
      # tar zxfv freeradius-1.0.0.tar.gz
      # cd freeradius-1.0.0
   ```

2. Configure, make and install:

   ```
   # ./configure
       # make
       # make install
   ```

   You can pass options to ***configure****. Use* ***./configure −−help*** *or read the README file, for more information.*

The binaries are installed in `/usr/local/bin` and `/usr/local/sbin`. The configuration files are found under `/usr/local/etc/raddb`.

If something went wrong, check the `INSTALL` and `README` included with the source. The RADIUS FAQ also contains valuable information.

3.2. Configuring FreeRADIUS

FreeRADIUS has a big and mighty configuration file. It's so big, it has been split into several smaller files that are just "included" into the main `radius.conf` file.

There is numerous ways of using and setting up FreeRADIUS to do what you want: i.e., fetch user information from LDAP, SQL, PDC, Kerberos, etc. In this document, user information from a plain text file, `users`, is used.

The configuration files are thoroughly commented, and, if that is not enough, the `doc/` folder that comes with the source contains additional information.

Configuring FreeRADIUS

1. The configuration files can be found under `/usr/local/etc/raddb/`

   ```
   # cd /usr/local/etc/raddb/
   ```

2. Open the main configuration file `radiusd.conf`, *and read the comments!* Inside the encrypted PEAP tunnel, an MS−CHAPv2 authentication mechanism is used.
   a. MPPE [RFC3078] is responsible for sending the PMK to the AP. Make sure the following settings are set:

```
# under MODULES, make sure mschap is uncommented!
mschap {
  # authtype value, if present, will be used
  # to overwrite (or add) Auth-Type during
  # authorization. Normally, should be MS-CHAP
  authtype = MS-CHAP

  # if use_mppe is not set to no, mschap will
  # add MS-CHAP-MPPE-Keys for MS-CHAPv1 and
  # MS-MPPE-Recv-Key/MS-MPPE-Send-Key for MS-CHAPv2
  #
  use_mppe = yes

  # if mppe is enabled, require_encryption makes
  # encryption moderate
  #
  require_encryption = yes

  # require_strong always requires 128 bit key
  # encryption
  #
  require_strong = yes

  authtype = MS-CHAP
  # The module can perform authentication itself, OR
  # use a Windows Domain Controller. See the radius.conf file
  # for how to do this.
}
```

b. Also make sure the "authorize" and "authenticate" contains:

```
authorize {
    preprocess
    mschap
    suffix
    eap
    files
}

authenticate {

     #
     #  MSCHAP authentication.
     Auth-Type MS-CHAP {
           mschap
      }

     #
     #  Allow EAP authentication.
     eap
 }
```

3. Then, change the `clients.conf` file to specify what network it's serving:

```
# Here, we specify which network we're serving
client 192.168.0.0/16 {
     # This is the shared secret between the Authenticator (the
     # access point) and the Authentication Server (RADIUS).
     secret          = SharedSecret99
     shortname       = testnet
 }
```

4. The `eap.conf` should also be pretty straightforward.

a. Set "default_eap_type" to "peap":

```
      default_eap_type = peap
```

b. Since PEAP is using TLS, the TLS section must contain:

```
    tls {
        # The private key password
        private_key_password = SecretKeyPass77
        # The private key
        private_key_file = ${raddbdir}/certs/cert-srv.pem
        #  Trusted Root CA list
        CA_file = ${raddbdir}/certs/demoCA/cacert.pem
        dh_file = ${raddbdir}/certs/dh
        random_file = /dev/urandom
        }
```

c. Find the "peap" section, and make sure it contain the following:

```
      peap {
        #  The tunneled EAP session needs a default
        #  EAP type, which is separate from the one for
        #  the non-tunneled EAP module.  Inside of the
        #  PEAP tunnel, we recommend using MS-CHAPv2,
        #  as that is the default type supported by
        #  Windows clients.
        default_eap_type = mschapv2
      }
```

5. The user information is stored in a plain text file `users`. A more sophisticated solution to store user information may be preferred (SQL, LDAP, PDC, etc.).

   Make sure the `users` file contains the following entry:

```
   "testuser"      User-Password == "Secret149"
```

4. Supplicant: Setting up Xsupplicant

The Supplicant is usually a laptop or other (wireless) device that requires authentication. Xsupplicant does the bidding of being the "Supplicant" part of the IEEE 802.1X−2001 standard.

4.1. Installing Xsupplicant

Installing Xsupplicant

1. Download the latest source from from http://www.open1x.org/

   ```
   # cd /usr/local/src
       # wget http://belnet.dl.sourceforge.net/sourceforge/open1x/xsupplicant-1.0.tar.gz
       # tar zxfv xsupplicant-1.0.tar.gz
       # cd xsupplicant
   ```
2. Configure, make, and install:

   ```
   # ./configure
       # make
       # make install
   ```
3. If the configuration file wasn't installed (copied) into the "etc" folder, do it manually:

   ```
   # mkdir -p /usr/local/etc/1x
       # cp etc/tls-example.conf /usr/local/etc/1x
   ```

If installation fails, check the `README` and `INSTALL` files included with the source. You may also check out the official documentation.

4.2. Configuring Xsupplicant

Configuring Xsupplicant

1. The Supplicant must have access to the root certificate.

   If the Supplicant needs to authenticate against the Authentication Server (authentication both ways), the Supplicant must have certificates as well.

   Create a certificate folder, and move the certificates into it:

   ```
   # mkdir -p /usr/local/etc/1x/certs
       # cp root.pem /usr/local/etc/1x/certs/
       # (copy optional client certificate(s) into the same folder)
   ```
2. Open and edit the configuration file:

   ```
   # startup_command: the command to run when Xsupplicant is first started.
   #   This command can do things such as configure the card to associate with
   #   the network properly.
   startup_command = <BEGIN_COMMAND>/usr/local/etc/1x/startup.sh<END_COMMAND>
   ```

   The `startup.sh` will be created shortly.
3. When the client is authenticated, it will transmit a DHCP request or manually set an IP address. Here,

the Supplicant sets its IP address manually in `startup2.sh`:

```
# first_auth_command: the command to run when Xsupplicant authenticates to
#   a wireless network for the first time.  This will usually be used to
#   start a DHCP client process.
#first_auth_command = <BEGIN_COMMAND>dhclient %i<END_COMMAND>
first_auth_command = <BEGIN_COMMAND>/usr/local/etc/1x/startup2.sh<END_COMMAND>
```

4. Since "−i" is just for debugging purpose (and may go away according to the developers), "allow_interfaces" must be set:

```
allow_interfaces = eth0
deny_interfaces = eth1
```

5. Next, under the "NETWORK SECTION", we'll configure PEAP:

```
# We'll be using PEAP
allow_types = eap_peap

# Don't want any eavesdropper to learn the username during the
# first phase (which is unencrypted), so 'identity hiding' is
# used (using a bogus username).
identity = <BEGIN_ID>anonymous<END_ID>

eap-peap {
   # As in tls, define either a root certificate or a directory
   # containing root certificates.
   root_cert = /usr/local/etc/1x/certs/root.pem
   #root_dir = /path/to/root/certificate/dir
   #crl_dir = /path/to/dir/with/crl
   chunk_size = 1398
   random_file = /dev/urandom
   #cncheck = myradius.radius.com   # Verify that the server certificate
                                    # has this value in its CN field.
   #cnexact = yes                   # Should it be an exact match?
   session_resume = yes

   # Currently 'all' is just mschapv2.
   # If no allow_types is defined, all is assumed.
   #allow_types = all # where all = MSCHAPv2, MD5, OTP, GTC, SIM
   allow_types = eap_mschapv2

   # Right now, you can do any of these methods in PEAP:
   eap-mschapv2 {
     username = <BEGIN_UNAME>testuser<END_UNAME>
     password = <BEGIN_PASS>Secret149<END_PASS>
   }
}
```

6. The Supplicant must first associate with the access point. The script `startup.sh` does that job. It is also the first command Xsupplicant executes.

Notice the bogus key we give to iwconfig (*enc 000000000*)! This key is used to tell the driver to run in encrypted mode. The key gets replaced after successful authentication. This can be set to *enc off* only if encryption is disabled in the AP (for testing purposes).

Both `startup.sh` and `startup2.sh` must be saved under `/usr/local/etc/1x/`.

```
#!/bin/bash
echo "Starting startup.sh"
# Take down interface (if it's up)
```

```
/sbin/ifconfig eth0 down
# To make sure the routes are flushed
sleep 1
# Configuring the interface with a bogus key
/sbin/iwconfig eth0 mode managed essid testnet enc 000000000
# Bring the interface up and make sure it listens to multicast packets
/sbin/ifconfig eth0 allmulti up
echo "Finished startup.sh"
```

7. This next file is used to set the IP address statically. This can be omitted if a DHCP server is present (as it typically is, in many access points).

```
#!/bin/bash
echo "Starting startup2.sh"
# Assigning an IP address
/sbin/ifconfig eth0 192.168.1.5 netmask 255.255.255.0
echo "Finished startup2.sh"
```

5. Authenticator: Setting up the Authenticator (Access Point)

During the authentication process, the Authenticator just relays all messages between the Supplicant and the Authentication Server (RADIUS). EAPOL is used between the Supplicant and the Authenticator; and, between the Authenticator and the Authentication Server, UDP is used.

5.1. Access Point

Many access point have support for 802.1X (and RADIUS) authentication. It must first be configured to use 802.1X authentication.

Configuring and setting up 802.1X on the AP may differ between vendors. Listed below are the required settings to make a Cisco AP350 work. Other settings to TIKP, CCMP etc. may also be configured.

The AP must set the ESSID to "testnet" and must activate:

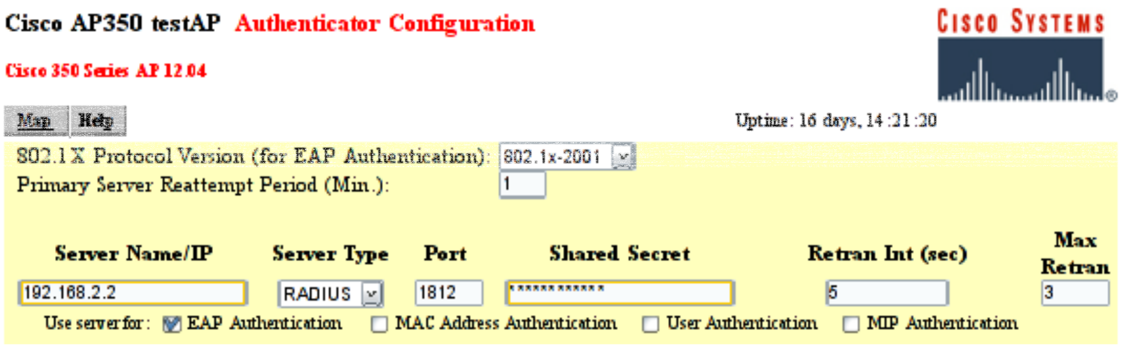

Figure AP350: The RADIUS configuration screen for a Cisco AP−350

- *802.1X−2001:* Make sure the 802.1X Protocol version is set to "802.1X−2001". Some older Access Points support only the draft version of the 802.1X standard (and may therefore not work).
- *RADIUS Server:* the name/IP address of the RADIUS server and the shared secret between the RADIUS server and the Access Point (which in this document is "SharedSecret99"). See figure AP350.
- *EAP Authentication:* The RADIUS server should be used for EAP authentication.

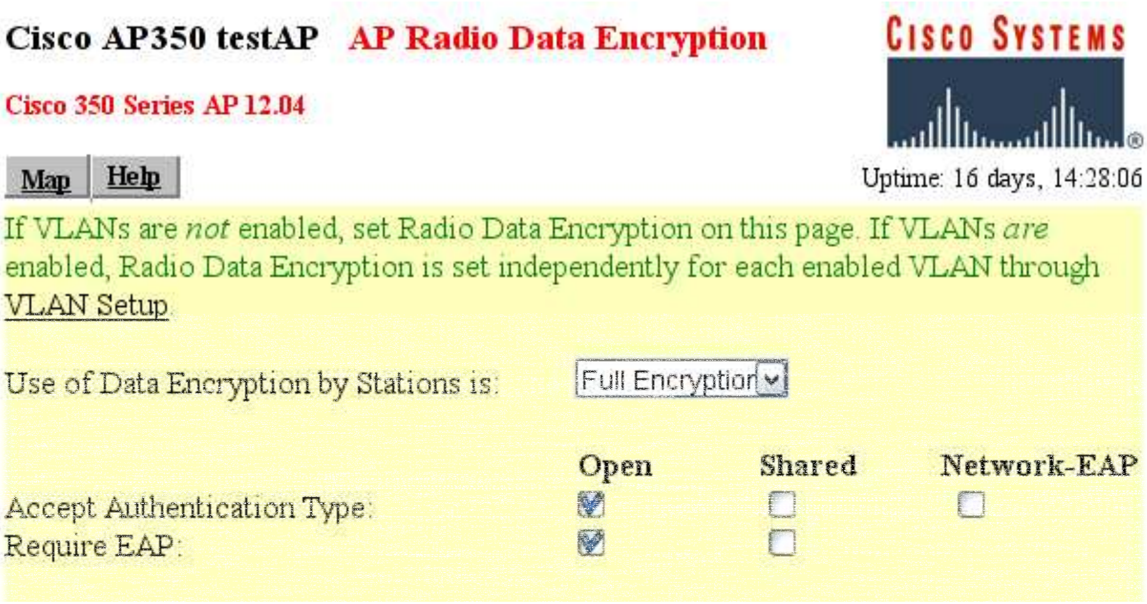

Figure AP350−2: The Encryption configuration screen for a Cisco AP−350

- *Full Encryption* to allow only encrypted traffic. Note that 802.1X may be used without using encryption, which is nice for test purposes.
- *Open Authentication* to make the Supplicant associate with the Access Point before encryption keys are available. Once the association is done, the Supplicant may start EAP authentication.
- *Require EAP* for the "Open Authentication". That will ensure that only authenticated users are allowed into the network.

5.2. Linux Authenticator

An ordinary Linux node can be set up to function as a wireless Access Point and Authenticator. How to set up and use Linux as an AP is beyond the scope of this document. Simon Anderson's Linux Wireless Access Point HOWTO may be of guidance.

6. Testbed

6.1. Testcase

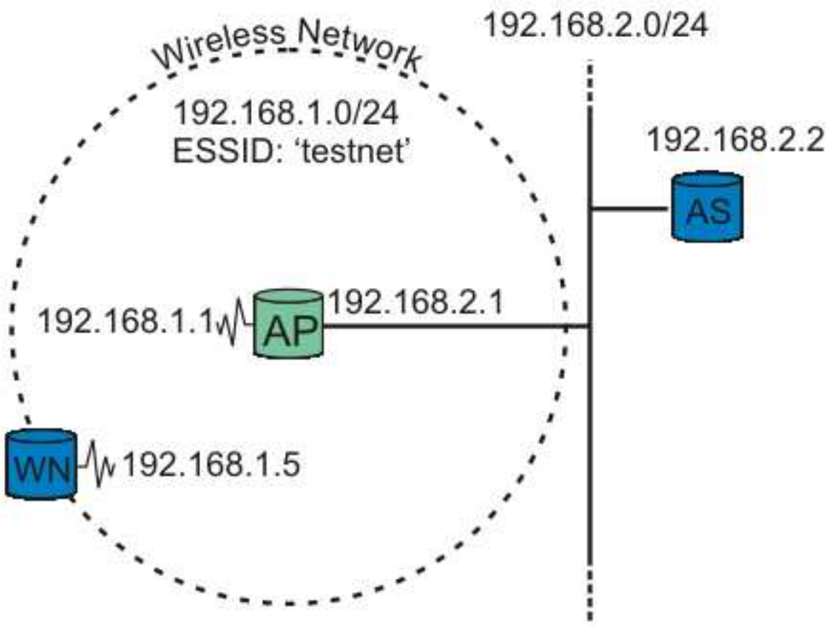

figure testbed: A wireless node request authentication.

Our testbed consists of two nodes and one Access Point (AP). One node functions as the Supplicant (WN), the other as the back–end Authentication Server running RADIUS (AS). The Access Point is the Authenticator. See figure testbed for explanation.

It is crucial that the Access Point be able to reach (ping) the Authentication Server, and vice versa!

6.2. Running some tests

Running some tests

1. The RADIUS server is started in debug mode. This produces *a lot* of debug information. The important snippets are below:

```
# radiusd -X
  Starting - reading configuration files ...
  reread_config:  reading radiusd.conf
  Config:   including file: /usr/local/etc/raddb/proxy.conf
  Config:   including file: /usr/local/etc/raddb/clients.conf
  Config:   including file: /usr/local/etc/raddb/snmp.conf
  Config:   including file: /usr/local/etc/raddb/eap.conf
  Config:   including file: /usr/local/etc/raddb/sql.conf
  ......
  Module: Loaded MS-CHAP
   mschap: use_mppe = yes
   mschap: require_encryption = no
```

```
  mschap: require_strong = no
  mschap: with_ntdomain_hack = no
  mschap: passwd = "(null)"
  mschap: authtype = "MS-CHAP"
  mschap: ntlm_auth = "(null)"
 Module: Instantiated mschap (mschap)
 ......
 Module: Loaded eap
  eap: default_eap_type = "peap" ❶
  eap: timer_expire = 60
  eap: ignore_unknown_eap_types = no
  eap: cisco_accounting_username_bug = no
 rlm_eap: Loaded and initialized type md5
  tls: rsa_key_exchange = no ❷
  tls: dh_key_exchange = yes
  tls: rsa_key_length = 512
  tls: dh_key_length = 512
  tls: verify_depth = 0
  tls: CA_path = "(null)"
  tls: pem_file_type = yes
  tls: private_key_file = "/usr/local/etc/raddb/certs/cert-srv.pem"
  tls: certificate_file = "/usr/local/etc/raddb/certs/cert-srv.pem"
  tls: CA_file = "/usr/local/etc/raddb/certs/demoCA/cacert.pem"
  tls: private_key_password = "SecretKeyPass77"
  tls: dh_file = "/usr/local/etc/raddb/certs/dh"
  tls: random_file = "/usr/local/etc/raddb/certs/random"
  tls: fragment_size = 1024
  tls: include_length = yes
  tls: check_crl = no
  tls: check_cert_cn = "(null)"
 rlm_eap: Loaded and initialized type tls
  peap: default_eap_type = "mschapv2" ❸
  peap: copy_request_to_tunnel = no
  peap: use_tunneled_reply = no
  peap: proxy_tunneled_request_as_eap = yes
 rlm_eap: Loaded and initialized type peap
  mschapv2: with_ntdomain_hack = no
 rlm_eap: Loaded and initialized type mschapv2
 Module: Instantiated eap (eap)
 ......
 Module: Loaded files
  files: usersfile = "/usr/local/etc/raddb/users" ❹
 ......
 Module: Instantiated radutmp (radutmp)
 Listening on authentication *:1812
 Listening on accounting *:1813
 Ready to process requests. ❺
```

❶ Default EAP type is set to PEAP.

❷ RADIUS's TLS settings are initiated here. The certificate type, location, and password are listet here.

❸ Inside the PEAP tunnel, MS−CHAPv2 is used.

❹ The username/password information is found in the `users` file.

❺ RADIUS server started successfully. Waiting for incoming requests.

The radius server is now ready to process requests!

The most interesting output is included above. If you get any error message instead of the last line, go over the configuration (above) carefully.

2. Now the Supplicant is ready to get authenticated. Start Xsupplicant in debug mode. Note that we'll see output produced by the two startup scripts: `startup.sh` and `startup2.sh`.

```
# xsupplicant -c /usr/local/etc/1x/1x.conf -i eth0 -d 6
  Starting /etc/1x/startup.sh
  Finished /etc/1x/startup.sh
  Starting /etc/1x/startup2.sh
  Finished /etc/1x/startup2.sh
```

3. At the same time, the RADIUS server is producing a lot of output. Key snippets are shown below:

```
  ......
  rlm_eap: Request found, released from the list
  rlm_eap: EAP/peap
  rlm_eap: processing type peap
  rlm_eap_peap: Authenticate
  rlm_eap_tls: processing TLS ❶
  eaptls_verify returned 7
  rlm_eap_tls: Done initial handshake
  eaptls_process returned 7
  rlm_eap_peap: EAPTLS_OK ❷
  rlm_eap_peap: Session established.  Decoding tunneled attributes.
  rlm_eap_peap: Received EAP-TLV response.
  rlm_eap_peap: Tunneled data is valid.
  rlm_eap_peap: Success
  rlm_eap: Freeing handler
  modcall[authenticate]: module "eap" returns ok for request 8
modcall: group authenticate returns ok for request 8
Login OK: [testuser/<no User-Password attribute>] (from client testnet port 37 cli 0002a56fa0
Sending Access-Accept of id 8 to 192.168.2.1:1032 ❸
        MS-MPPE-Recv-Key = 0xf21757b96f52ddaefe084c343778d0082c2c8e12ce18ae10a79c550ae❹1a5206
        MS-MPPE-Send-Key = 0x5e1321e06a45f7ac9f78fb9d398cab5556bff6c9d003cdf8161683bfb7e7af18
        EAP-Message = 0x030a0004
        Message-Authenticator = 0x00000000000000000000000000000000
        User-Name = "testuser"
```

❶ TLS session startup. Doing TLS−handshake.

❷ The TLS session (PEAP−encrypted tunnel) is up.

❸ The Supplicant has been authenticated successfully by the RADIUS server. An "Access−Accept" message is sent.

❹ The *MS−MPPE−Recv−Key* [RFC2548 section 2.4.3] contains the Pairwise Master Key (PMK) destined to the Authenticator (access point), encrypted with the MPPE Protocol [RFC3078], using the shared secret between the Authenticator and Authentication Server as key. The Supplicant derives the same PMK from MK, as described in Key Management.

4. The Authenticator (access point) may also show something like this in its log:

```
  00:02:16 (Info): Station 0002a56fa08a Associated
  00:02:17 (Info): Station=0002a56fa08a User="testuser" EAP-Authenticated
```

That's it! The Supplicant is now authenticated to use the Access Point!

7. Note about driver support and Xsupplicant

As described in Key Management, one of the big advantages of using Dynamic WEP/802.11i with 802.1X is the support for session keys. A new encryption key is generated for each session.

Xsupplicant only supports "Dynamic WEP" as of this writing. Support for WPA and RSN/WPA2 (802.11i) is being worked on, and is estimated to be supported at the end of the year/early next year (2004/2005), according to Chris Hessing (one of the Xsupplicants developers).

Not all wireless drives support dynamic WEP, nor WPA. To use RSN (WPA2), new support in hardware may even be required. Many older drivers assume only one WEP key will be used on the network at any time. The card is reset whenever the key is changed to let the new key take effect. This triggers a new authentication, and there is a never−ending loop.

At the time of writing, most of the wireless drivers in the base Linux kernel require patching to make dynamic WEP/WPA work. They will, in time, be upgraded to support these new features. Many drivers developed outside the kernel, however, support for dynamic WEP; HostAP, madwifi, Orinoco, and atmel should work without problems.

Instead of using Xsupplicant, wpa_supplicant may be used. It has support for both WPA and RSN (WPA2), and a wide range of EAP authentication methods.

8. FAQ

Do not forget to check out the FAQ section of both the FreeRADIUS (highly recommended!) and Xsupplicant Web sites!

8.1. Is it possible to allow user−specific Xsupplicant configuration, to avoid having a global configuration file?

No, not at the moment.

8.2. I don't want to use PEAP; can I use EAP−TTLS or EAP−TLS instead?

Yes. To use EAP−TTLS, only small changes to the configuration used in this document are required. To use EAP−TLS, client certificates must be used as well.

8.3. Can I use a Windows Supplicant (client) instead of GNU/Linux?

Yes. Windows XP SP1/Windows 2000 SP3 has support for PEAP MSCHAPv2 (used in this document). A Windows HOWTO can be found here: FreeRADIUS/WinXP Authentication Setup

8.4. Can I use a Active Directory to authenticate users?

Yes. FreeRADIUS can authenticate users from AD by using "ntlm_auth".

8.5. Is there any Windows Supplicant clients available?

Yes. As of Windows XP SP1 or Windows 2000 SP3, support for WPA (PEAP/MS−CHAPv2) is supported. Other clients include (not tested) Secure W2 (free for non−commercial) and WIRE1X. Funk Software also has a commercial client available.

---

9. Useful Resources

Only IEEE standards older than 12 months are available to the public in general (through the "Get IEEE 802 Program"). So the new *802.11i* and *802.1X−2004* standards documents are not available. You must be a IEEE participant to get hold of any drafts/work in progress papers (which actually isn't that hard − just join a mailing list and say you are interested).

1. FreeRADIUS Server Project http://www.freeradius.org/
2. Open1x: Open Source implementation of IEEE 802.1X (Xsupplicant) http://www.open1x.org/
3. The Open1x User's Guide http://sourceforge.net/docman/display_doc.php?docid=23371&group_id=60236
4. Port−Based Network Access Control (802.1X−2001) http://standards.ieee.org/getieee802/download/802.1X−2001.pdf
5. RFC2246: The TLS Protocol Version 1.0 http://www.ietf.org/rfc/rfc2246.txt
6. RFC2459: Internet X.509 Public Key Infrastructure − Certificate and CRL Profile http://www.ietf.org/rfc/rfc2459.txt
7. RFC2548: Microsoft Vendor−specific RADIUS Attributes http://www.ietf.org/rfc/rfc2548.txt
8. RFC2716: PPP EAP TLS Authentication Protocol http://www.ietf.org/rfc/rfc2716.txt
9. RFC2865: Remote Authentication Dial−In User Service (RADIUS) http://www.ietf.org/rfc/rfc2865.txt
10. RFC3079: Deriving Keys for use with Microsoft Point−to−Point Encryption (MPPE) http://www.ietf.org/rfc/rfc3079.txt
11. RFC3579: RADIUS Support For EAP http://www.ietf.org/rfc/rfc3579.txt
12. RFC3580: IEEE 802.1X RADIUS Usage Guidelines http://www.ietf.org/rfc/rfc3580.txt
13. RFC3588: Diameter Base Protocol http://www.ietf.org/rfc/rfc3588.txt
14. RFC3610: Counter with CBC−MAC (CCM) http://www.ietf.org/rfc/rfc3610.txt
15. RFC3748: Extensible Authentication Protocol (EAP) http://www.ietf.org/rfc/rfc3748.txt
16. Linux Wireless Access Point HOWTO http://oob.freeshell.org/nzwireless/LWAP−HOWTO.html
17. SSL Certificates HOWTO http://www.tldp.org/HOWTO/SSL−Certificates−HOWTO/
18. OpenSSL: x509(1) http://www.openssl.org/docs/apps/x509.html

10. Copyright, acknowledgments and miscellaneous

10.1. Copyright and License

10.2. How this document was produced

This document was written in DocBook XML using Emacs.

10.3. Feedback

Suggestions, corrections, additions wanted. Contributors wanted and acknowledged. Flames not wanted.

I can always be reached at `<lars strand at gnist org>`

Homepage: http://www.gnist.org/~lars/

10.4. Acknowledgments

Thanks to Andreas Hafslund `<andreha at unik no>` and Thales Communication for initial support.

Also thanks to Artur Hecker `<hecker at enst fr>`, Chris Hessing `<chris hessing at utah edu>`, Jouni Malinen `<jkmaline at cc hut fi>` and Terry Simons `<galimore at mac com>` for valuable feedback!

Thanks to Rick Moen `<rick at linuxmafia com>` for doing a language review!

A. GNU Free Documentation License

Version 1.2, November 2002

A.1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world−wide, royalty−free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front−matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front−Cover Texts or Back−Cover Texts, in the notice that says that the Document is released under this License. A Front−Cover Text may be at most 5 words, and a Back−Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine−readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard−conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine−generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front−Cover Texts on the front cover, and Back−Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine−readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer−network location from which the general network−using public has access to download using public−standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front−matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties−−for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front−Cover Text, and a passage of up to 25 words as a Back−Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front−Cover Text and one of Back−Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

A.7. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12. ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front−Cover Texts, and no Back−Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front−Cover Texts and Back−Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front−Cover Texts being LIST, and with the Back−Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.