Avaya 3631 Wi-Fi IP Phone Wireless Security Configuration Note

This is a configuration note about how to enable and setup various wireless security features supported by the Avaya 3631 IP phone.

TABLE OF CONTENTS

1 Introduction 4
1.1 Security Features Supported by Avaya 3631 IP phone 4
1.2 Notation Used 4
2 WEP Setup Guide 5
2.1 Wireless Access Point Configuration 5
2.2 Phone Configuration 5
3 WPA-PSK/WPA2-PSK Setup Guide 6
3.1 Wireless Access Point Configuration 6
3.2 Phone Configuration 6
4 WPA-802.1x/WPA2-802.1x Setup Guide 7
4.1 Phone Configuration 7
4.1.1 TLS 7
4.1.2 LEAP 8
4.1.3 PEAP-GTC 9
4.1.4 PEAP-MSCHAPv2 9
4.1.5 TTLS-CHAP 10
4.1.6 TTLS-MD5 10
4.1.7 TTLS-MSCHAPv1 11
4.1.8 TTLS-MSCHAPv2 11
4.2 Generating Phone Certificates and Keys 12
4.2.1 Using Microsoft IAS 12
4.2.2 Using FreeRADIUS 13
4.3 Server Configuration 15
4.3.1 Microsoft Internet Authentication Server (IAS) 16
4.4 Access Point Configuration for 802.1x based authentication 17
4.4.1 Cisco Aironet 1200 series 17
4.4.2 Meru Access Points 18
5 Bulk Configuration 19
5.1.1 Via USB 19
6 Appendix 20
6.1 OpenSSL Certificate Creation 20
6.1.1 File: CA_CONFIG 20
6.1.2 File: CA.root 20
6.1.3 File: CA.svr 21
6.1.4 File: CA.clt 22
6.1.5 File: xpextensions 22
6.2 Microsoft IIS for IAS Configuration 22

6.3 Using OpenSSL to convert certificates.....23

1 Introduction

The Avaya 3631 is an 802.11 b/g Wi-Fi standards-based wireless IP phone. The 3631 uses H.323 standards to communicate with an Avaya Communication Manager.

1.1 Security Features Supported by Avaya 3631 IP phone

The 3631 Wi-Fi telephone provides support for the following security features:

- Wired Equivalent Privacy (WEP) 64-bit and 128-bit encryption.
- Wi-Fi Protected Access (WPA) with Pre-Shared Key (PSK) authentication and TKIP encryption.
- Wi-Fi Protected Access 2 (WPA2) with Pre-Shared Key (PSK) authentication and AES encryption.
- Wi-Fi Protected Access (WPA) with 802.1X authentication and TKIP encryption.
- Wi-Fi Protected Access 2 (WPA2) with 802.1X authentication and AES encryption.

The EAP methods supported for 802.1x authentication are:

- TLS
- LEAP
- PEAP-GTC
- PEAP-MSCHAPv2
- TTLS-CHAP
- TTLS-MD5
- TTLS-MSCHAP
- TTLS-MSCHAPv2

The customer can configure up to three separate profiles on the Wi-Fi telephone (with independent security features) either via the phone's display interface or via the bulk configuration utility. Both manners of configuration are described within this document.

1.2 Notation Used

AP	Access Point
WLAN	Wireless Local Area Network
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
EAP	Extensible Authentication Protocol
VLAN	Virtual Local Area Network
SSID	Service Set Identifier
PSK	Pre-Shared Key
IAS	Internet Authentication Service (Microsoft Windows Server service)
IIS	Internet Information Services (Microsoft Windows Server service)
TKIP	Temporal Key Integrity Protocol
AES	Advanced Encryption Standard
WMM	Wi-Fi Multimedia

2 WEP Setup Guide

2.1 Wireless Access Point Configuration

Almost all access points support WEP encryption. WEP setup guides for specific hardware should be readily available online.

2.2 Phone Configuration

From the Advanced settings menu, choose Admin access. Enter the appropriate Admin password and choose Access profiles, then select one of the three available profiles to be configured.

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect WEP security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect WEP security.
- **Security type**: WEP
- **Encryption type**: If a 64-bit key was used, select WEP-40/64. If a 128-bit key was used, select WEP-104/128.
- **Encryption key**: If the encryption key was entered as a hexadecimal sequence, choose to enter the key in Hex mode when prompted. Otherwise, if the key was entered as an ASCII string, enter ASCII mode. Then enter the full WEP encryption key as set during the configuration of the access point.
- **EAP type**: Disable
- **EAP identity/username/password**: With EAP type set to *Disable*, these values have no effect.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the Phone IP address, Subnet mask, Default gateway, DNS servers and Domain fields.

Having set these values correctly, save the settings and restart the phone from the main Options menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars) and DHCP may occur (depending on your above settings).

3 WPA-PSK/WPA2-PSK Setup Guide

3.1 Wireless Access Point Configuration

Most modern access points support WPA encryption with a Pre-Shared Key, and many also provide WPA2 support. WPA/WPA2 setup guides for specific hardware should be readily available online.

Note that WPA uses TKIP for encryption; WPA2 uses AES for encryption.

3.2 Phone Configuration

From the *Advanced settings* menu, choose *Admin* access. Enter the appropriate *Admin* password and choose *Access profiles*, then select one of the three available profiles to be configured.

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect WPA/WPA2 security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect WPA/WPA2 security.
- **Security type**: If WPA-PSK (TKIP encryption) is being used on the access point, choose *WPA-PSK*. If WPA2-PSK (AES encryption) is being used on the access point, choose *WPA2-PSK*.
- **Encryption type**: If WPA-PSK was selected above, choose *TKIP* encryption. If WPA2-PSK was selected above, choose *AES* encryption.
- **Encryption key**: If the encryption key was entered as a hexadecimal sequence, choose to enter the key in *Hex* mode when prompted. Otherwise, if the key was entered as an ASCII string, enter *ASCII* mode. Then enter the full Pre-Shared Key (PSK) encryption key as set during the configuration of the access point.
- **EAP type**: *Disable*
- **EAP identity/username/password**: With *EAP type* set to *Disable*, these values have no effect.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars) and DHCP may occur (depending on your above settings).

4 WPA-802.1x/WPA2-802.1x Setup Guide

Avaya 3631 Wi-Fi IP phone supports the following WPA(2)-Enterprise methods of authentication and encryption:

- Wi-Fi Protected Access (WPA) with 802.1X authentication and TKIP encryption.
- Wi-Fi Protected Access 2 (WPA2) with 802.1X authentication and AES encryption.

The EAP methods supported for 802.1x authentication are:

- TLS
- LEAP
- PEAP-GTC
- PEAP-MSCHAPv2
- TTLS-CHAP
- TTLS-MD5
- TTLS-MSCHAP
- TTLS-MSCHAPv2

The Avaya 3631 Wi-Fi phone comes from the factory with no certificates installed. According to customer requirements, certificates may need to be installed on the phone that will be used in the authentication process. Certificates can be installed on the phone using the USB interface or over the air using the 46xxsettings file. Both options are discussed in Section 5.

3631 phones have no Public Key Infrastructure (PKI) client like SCEP/PKCS, do not generate any certificates internally that contain the public key or private keys, and cannot request a certificate authority (CA) to sign the phone certificate. As such, depending on the chosen EAP mechanism, certificates may need to be manually provisioned on the 3631 IP phone.

The certificates uploaded on the phone must be named as below.

1. **private_key.pem** - Phone's private certificate which contains the private key.
2. **user_cert.pem** - Phone's public certificate that has been signed by a trusted CA server.
3. **cacert*x*.pem** - CA certificates which contain the trusted CA authority's public key. One certificate per profile can be provisioned on the phone. This should be named as *cacertx.pem*, where x is the profile number (e.g. *cacert1.pem* for the first profile).
4. **private_key_passwd.txt** - If the *private_key.pem* is encrypted using a password while it is created using a certificate generation tool, then the password file also need to be provisioned on the phone to decrypt *private_key.pem*.

The particular certificates and files that are required to be provisioned on the phone depend on the EAP types you choose for the network. For instance, all supported EAP types (with the exception of EAP-TLS) require only the *cacertx.pem* certificate to be present on the phone. For EAP-TLS, *private_key.pem*, *user_cert.pem*, *cacertx.pem* and *private_key_passwd.txt* files are also required.

These files can be uploaded via Bulk Configuration (see Bulk Configuration section), or via USB and phone menus. See Section 5 for details.

Note the naming of the `cacertx.pem` file – rename it as appropriate to the access profile number with which this trusted root CA certificate is to be used. For example, for use with the first access profile, name the file `cacert1.pem`.

Once the files are uploaded, follow the following procedure on the phone to configure various EAP types.

4.1 Phone Configuration

4.1.1 TLS

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point

- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *TLS*
- **EAP identity**: Normally the phone's installed client certificate's Common Name (see Appendix 6.1).
- **EAP username**: Not required.
- **EAP password**: Not required.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP may occur (depending on your above settings).

Note that for TLS, all these certificate files private_key.pem, user_cert.pem, cacertx.pem and private_key_passwd.txt files are required.

4.1.2 LEAP

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *LEAP*
- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.
- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP negotiation may occur (depending on your above settings).

4.1.3 PEAP-GTC

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *PEAP-GTC*
- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.
- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP may occur (depending on your above settings).

4.1.4 PEAP-MSCHAPv2

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *PEAP-MSCHAPV2*
- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.
- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP negotiation may occur (depending on your above settings).

4.1.5 TTLS-CHAP

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *TTLS-CHAP*
- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.
- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP may occur (depending on your above settings).

4.1.6 TTLS-MD5

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *TTLS-MD5*
- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.

- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP negotiation may occur (depending on your above settings).

4.1.7 TTLS-MSCHAPv1

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *TTLS-MSCHAP*
- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.
- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP may occur (depending on your above settings).

4.1.8 TTLS-MSCHAPv2

From the profile configuration scheme, the following options should be set:

- **Profile name**: Set to a meaningful value
- **SSID**: As set on the wireless access point
- **WMM mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Power save mode**: Depends on the capabilities of your access point – should not affect 802.1x security.
- **Security type**: During access point configuration, if WPA-Enterprise or TKIP Cipher was configured on the AP, set the phone to use *WPA-802.1x*. Otherwise, if WPA2-Enterprise or AES Cipher was configured on the AP, set the phone to use *WPA2-802.1x*.
- **Encryption type**: If *WPA-802.1x* was set above, use *TKIP*. If *WPA2-802.1x* was set above, use *AES*.
- **Encryption key**: With the phone configured to use 802.1x security, this value has no effect.
- **EAP type**: *TTLS-MSCHAPV2*

- **EAP identity**: The user's unique username that will be used to authenticate with the RADIUS login server (normally the same as the EAP username below). Sometimes known as the "Remote Access Username".
- **EAP username**: The user's username that will be used to authenticate with the RADIUS login server.
- **EAP password**: The user's password that will be used in conjunction with the above username when authenticating with the RADIUS login server.
- **Use DHCP**: Speak to your network administrator as to whether your site uses DHCP for IP address assignment. If not, you will need to manually specify the **Phone IP address**, **Subnet mask**, **Default gateway**, **DNS servers** and **Domain** fields.

Having set these values correctly, save the settings and restart the phone from the main *Options* menu. Once the phone has restarted, it should connect to the AP (the signal strength meter will show a number of bars), 802.1x authentication will proceed and complete, and DHCP may occur (depending on your above settings).

4.2 Generating Phone Certificates and Keys

This section describes how to generate the necessary certificates required by the 3631 phone. All supported EAP types, with the exception of EAP-TLS, only require the *cacertx.pem*[1] certificate to be available on the phone. When using the EAP-TLS type, *private_key.pem*, *user_cert.pem*, *cacertx.pem* and *private_key_passwd.txt* files are also required.

Note that the 3631 phone supports only the *.pem* certificate file format. If you have an existing CA certificate in DER format (as can be generated using Microsoft IAS and generally ending with a file extension such as *.cer*, *.crt* or *.der*), you are not be able to simply rename the file extension from *.cer* to *.pem*. Instead, if the certificate is in DER format, you must follow the PEM conversion procedure outlined in section 6.3. Once the customer has trusted CA certificates in X.509 format, rename these certificates to *cacertx.pem* and transfer the certificates to the phone using USB or over the air as described earlier. Note that Microsoft IAS can be told to export in X.509 format and will still generate a file with extension *.cer*; in this case it is safe to simply rename the file (this is discussed further in section 4.2.1).

If EAP-TLS is required, the user's private key certificate file and public key certificate file need to be generated and transferred to the 3631 phone. If a password was used to protect the private key file, then the password file also needs to be transferred to the phone over USB or over the air as described previously. The files need to be named as *private_key.pem, user_cert.pem* and *private_key_passwd.txt*.

Customers may obtain a *cacertx.pem* file from a commercial certification authority or maintain their own CA servers and issue certificates. This section below describes how to generate the required certificates using their own CA servers. If you have a CA certificate server running and already have a certificate, you need to export the CA cert to the phone, perform any necessary conversion to get it into *.pem* format (see section 6.3), and rename it as *cacertx.pem*.

4.2.1 Using Microsoft IAS

Microsoft Windows Server 2003 (service pack 2) was used in the production of this guide.

First, confirm that IIS and IAS have been correctly setup to work together by referring to Section 6.2. Once this is done, on your Microsoft IAS server go to *Administrative Tools→Certification Authority*. Expand the *Certificate Authority* node and right click on your listed local server, and select *Properties*. Click the *View Certificate* button in the *General* tab and go to the *Details* tab. Click *Copy to file.* Select *Base-64 encoded X.509* format. Choose a local location to store the CA certificate file as *cacertx.cer*. Then select *Finish*. Rename this exported CA certificate from *cacertx.cer* to *cacertx.pem* ready for use on the phone[2].

If you are using EAP-TLS mechanism, then certificates now need to be created for the phone authentication stages. The procedure described below will create *private_key.pem, user_cert.pem* and *private_key_passwd.txt* as required for use with EAP-TLS.

---

[1] Throughout this document, replace the *x* in *cacertx.pem* with 1, 2 or 3 depending on the profile number on the phone with which this CA certificate will be associated.

[2] Note that, since the X.509 format was selected for export, you do not need to perform any of conversion steps listed in section 6.3.

To generate these certificates and keys, we need to connect to the Microsoft IAS server and request these certificates be generate. To do this, we connect to the IAS server's Internet Information Services (IIS) server and fill out an online request form. Open Microsoft Internet Explorer and navigate to:

```
http://<publicIP>:<port>/CertSrv/
```

Replace *publicIP* with the public IP address of your Microsoft IAS server (which is running IIS to handle certificate generation requests). Replace *port* with the value used by your IIS server. If this fails, please refer to Section 6.2.

The "Microsoft Certificate Services" page should be displayed. Select the "Request a certificate" task, then "Advanced certificate request" and "Create and submit a request to this CA". Fill in the "Identifying Information" form (parts can be left blank), ensuring that the "Name" is set to the username that the client uses to authenticate to IAS with. Select a "Client Authentication Certificate" as the "Type of Certificate" needed. In the "Key Options" part, select:

- Create new key set
- CSP: Microsoft Enhanced Cryptographic Provider
- Key Usage: Both
- Key Size: 1024
- Automatic key container name
- Check the "Mark keys as exportable" option

Finally hit Submit. The certificate request will be placed into the pending queue. You will now need to issue the certificate: on the Microsoft IAS server itself, go to *Administrative Tools→Certification Authority* and navigate to *Pending Certificates*. Right click on the newly created pending certificate and select *Issue*. Finally, use Internet Explorer to return to *http://<publicIP>:<port>/CertSrv/* on the same computer that issued the certificate request. Then select *View the status of a pending certificate request*. Finally select the newly issued certificate and add it ready for use.

Once added, this client certificate is placed in Microsoft Internet Explorer's certificate cache. You will then need to extract it for use by the phone. To do this, click *Tools→Internet Options* in Internet Explorer. Then select the *Content* tab and choose *Certificates*. The newly created certificate should be listed in the Personal certificates directory. Select this certificate that will be used with the phone, and click *Export...* When prompted, choose to export the private key. Select the *Personal Information Exchange* format, *unchecking* strong protection and all other options. After clicking next, enter a password to use for encrypting the private key file – remember this password for future use. Finally choose a filename (ending in the *.pfx* extension, e.g. *username.pfx*) to save the exported certificate and key to.

The exported file needs to be converted into a format ready for use by the 3631phone. To do this, first rename this exported file to end in the *.p12* extension (e.g. *username.p12*). Then, using OpenSSL[3] tools, execute the following commands to extract separate private key and client certificates from *<export_file>.p12*:

```
# openssl pkcs12 -in <export_file>.p12 -clcerts -nokeys –out
user_cert.pem
# openssl pkcs12 -in export<export_file>.p12 -nocerts -out
private_key.pem
```

Enter a private key password when prompted. This will complete with the creation of the two files, `user_cert.pem and private_key.pem`

4.2.2 Using FreeRADIUS

FreeRADIUS is an open source RADIUS implementation. There are implementations available to run on both Microsoft Windows and Linux:

- For Linux, www.freeradius.org
- For Windows, www.freeradius.net

FreeRADIUS version 1.1.3 was used in the production of this guide.

[3] Available from http://www.openssl.org, specifically http://www.openssl.org/related/binaries.html

For MS Windows: Note that once FreeRADIUS is installed, you will need to tell Microsoft Windows to start this application as a service. To do this, execute the following command from the Command Prompt (adjust as appropriate to your install location):

```
C:\Program Files\FreeRADIUS.net\bin>cygrunsrv -I "FreeRADIUS" -
-path "radiusd.exe" --args "-f -d ../etc/raddb" -n -c
"C:\Program Files\FreeRADIUS.net\bin"
```

Certificates are required for both server and client. Whilst there are a number of methods available for the generation of certificates for use by client/server, a short guide based around the use of the free tool OpenSSL is provided below in *OpenSSL Certificate Creation*.

A set of default configuration files are provided with a standard install of FreeRADIUS. Some slight configuration is required in order to allow the RADIUS server to be used in 802.1x authentication exchanges. Whenever a set of configuration changes are made, be sure to restart the RADIUS server for the changes to take effect. The appropriate contents for these configuration files (normally found in your FreeRADIUS install directory) are discussed below.

4.2.2.1 radiusd.conf

The most important settings should be confirmed to be set as below (check these lines are not commented out):

```
...
port = 1812
...
certsdir = ${sysconfdir}/raddb/certs/my_certs
...
modules {
      ...
      mschap {
            ...
      }
      ...
      $INCLUDE ${confdir}/eap.conf
      ...
}
authorize {
      ...
      mschap
      ...
      eap
      ...
      files
      ...
}
authenticate {
      ...
      Auth-Type MS-CHAP {
            mschap
      }
      ...
      eap
      ...
}
```

4.2.2.2 eap.conf

Confirm that the EAP settings file is correctly configured.

```
eap {
      ...
      default_eap_type = peap
      ...
      tls {
            private_key_password = <private key password>
            private_key_file = ${certsdir}/<private key file>
            certificate_file = ${certsdir}/<serv_cert_file>
```

```
            CA_file = ${certsdir}/<root_CA_file>
        }
        ...
        gtc {
            ...
            auth_type = Local
            ...
        }
        ...
        peap {
            ...
            default_eap_type = mschapv2
            ...
        }
        ...
    }
```

The *private_key_file* should be set to the server's private key, and the *certificate_file* set to the server's certificate. If OpenSSL was used (as described in this document) for certificate generation, both these fields will be set to **`<server_name>.pem`**. The private key password should be set to the string used to encrypt the server's private key. Finally, the CA_file should be set to the trusted root Certificate Authority list file (**`root.pem`** in the OpenSSL example).

4.2.2.3 clients.conf

Add an entry allowing access for the IP address of the wireless Access Point:

```
client <IP>/<subnet> {
        secret     = <shared_secret>
        shortname  = <any_name>
}
```

The IP/subnet can be specified as a single IP address ("192.168.1.23") or a subnet ("192.168.1.0/24"). Only IP addresses in these ranges are able to attempt to connect to the RADIUS server. Furthermore, the wireless access point must use the same correct *shared_secret* value.

4.2.2.4 users.conf

In the case of this simple setup, authentication credentials for users will be added directly into this file. To do this, add an entry similar to that below for users that will be able to login via 802.1x based authentication:

```
<username>      User-Password == "<password>"
                Fall-Through = Yes
```

For example:

```
Fred            User-Password == "mypassword"
                Fall-Through = Yes

Bob             User-Password == "anotherpasswd"
                Fall-Through = Yes

# etc. for each user
```

For more advanced configuration (including integrating with the authentication of a Windows domain or UNIX NIS login system), please see the FreeRADIUS documentation. Also note that, if a client certificate authentication scheme is being used (for example, with EAP-TLS), then the client's certificate Common Name must be the same as the RADIUS username set above.

4.3 Server Configuration

A RADIUS server is used in 802.1x solutions in order to authenticate client logins and passwords – it acts as an authentication server. The section below are intentionally simple and do not cover more complex setups such as the use of enterprise domain logins.

4.3.1 Microsoft Internet Authentication Server (IAS)

If you do not already have user accounts setup, go to *Administrative Tools→Computer Management*. Navigate to "Users", and create user accounts that will be used by 3631 IP phone wireless users. Set the password to a valid value, and ensure that the "Password never expires" box is checked.

For those new users you just created, and for existing users already existing on the system, you will need to enable "Remote Access Permission" for them to be able to authenticate via 802.1x. To do this, go to *Administrative Tools→Computer Management* and expand the left hand tree *System Tools→Local Users and Groups→Users*. For each user listed in the "Users" list that you wish to be able to authenticate using 802.1x, you will need to right-click, Properties and go to the "Dial-in" tab. Select "Allow access" in the "Remote Access Permission (Dial-in or VPN)" section.

Now expand the *Computer Management (local)* node, and navigate to "Groups". Create a new group say for example "Wireless Users" and add those users you wish to be able to authenticate to this new group.

4.3.1.1 IAS Generating Server Certificates

First, confirm that IIS and IAS have been correctly setup to work together by referring to Section 6.2. Then go to *Administrative Tools→Certification Authority*. Expand the *Certificate Authority* node and right click on your listed local server, and select Start if not already started. When you start *Certificate Services*, you may be prompted to create a root certificate. Ensure you define this new certificate to be a root certificate capable of being used to create other certificates.

You now need to create a *Server Authentication* certificate for use by PEAP (an authentication mechanism that can be supported by 802.1x). To do this, you will need to access the Certificate Authority via its online web page.

With IIS now setup, open Microsoft Internet Explorer and navigate to:

`http://<publicIP>:<port>/CertSrv/`

Replace *publicIP* with the public IP address of your Windows server. Replace *port* with the value recorded above (as specified in IIS Manager). If this fails, please refer to Section 6.2.

A Microsoft Certificate Services page should be displayed. Select the *Request a certificate* link, then *advanced certificate request* link and *Create and submit a request to this CA*. Fill in the "Identifying Information" form (parts can be left blank). Select a "Server Authentication Certificate" as the Type of Certificate needed. In the Key Options part, select:

- Create new key set
- **CSP**: Microsoft DH SChannel Cryptographic Provider
- **Key Usage**: Signature
- **Key Size**: 1024
- Automatic key container name
- Check the "Store Certificate in the local computer certificate store" option

Finally hit Submit. The certificate request will be placed into the pending queue. You will now need to issue the certificate: go to *Administrative Tools→Certification Authority* and navigate to "Pending Certificates". Right click on the newly created pending certificate and select "Issue". Finally, return to *http://<publicIP>:<port>/CertSrv/* in the web browser. Then select "View the status of a pending…". Finally select the newly issued certificate and add it ready for use.

4.3.1.2 Authentication Service

Go to *Administrative Tools→Internet Authentication Service*. Start *Internet Authentication Service* by right-clicking and selecting "start". Right click again on *IAS (local)* and select "Properties". Ensure that the Ports list contains 1645 in Authentication, and 1646 in Accounting.

Now right-click on the RADIUS Clients folder and select "New RADIUS Client". Enter in a name for the client in the "Friendly name" field, along with the IP address of the wireless AP in the "Client address" field. Click Next and select "RADIUS Standard" as the Client-Vendor, and set the Shared Secret to secure value (record this value as it will be needed later when configuring the access point).

To help in debugging, go to "Remote Access Logging" and right click select "Properties" on the "Local File" method. Tick all the logging options on the Settings tab, and set the location of the log file.

Now right-click on the "Remote Access Policies" folder, and select "New Remote Access Policy". Give the policy a meaningful name, and select "Use the wizard…". Select Next and choose the Wireless method of access. Choose Next again, and choose "Group" and "Add..." the "Wireless Users" group which you created initially on the server. Hit Next, and choose "Protected EAP (PEAP)" then Configure. Ensure that the certificate you created previously is selected and able to be used. Finally hit Next and Finish. Right click on the newly created Access Policy, and choose Properties. Remove the first "NAS-Port-Type" rule leaving just the "Windows-Groups" rule. Choose "Edit Profile…". On the Encryption tab, tick all the encryption types. On the Authentication tab, enable "MS-CHAP v2", "MS-CHAP" and "CHAP". Click EAP Methods and confirm PEAP is listed as an EAP type. Select PEAP, and choose "Edit…", confirming that the created certificate is selected. Finally hit OK in all the open dialog boxes to return back to the main Internet Authentication Service page.

Expand the "Connection Request Processing" folder, and right-click on the "Connection Request Policies" item. Choose "New Connection Request Policy" and choose Next. Select "A custom policy" and name the policy name something memorable. Hit Next and choose Add… to add a new Policy Condition. Choose "Client-IP-Address", hit "Add…" and then type the IP address of your wireless AP. Finally hit OK to close this dialog, then Next back in the Wizard until you can hit Finish to create the policy. Right-click on the newly created policy and select Properties. Choose "Edit Profile…" and ensure that "Authenticate requests on this server" is selected on the Authentication tab. Hit OK and return back to the main Internet Authentication Service page.

4.3.1.3 Debugging Authentication

If you are having trouble with authentication, install Wireshark and confirm RADIUS messages are being transmitted from AP to server. Also, install the "Windows Support Tools" package[4], and use the isaparse.exe tool to inspect the IAS server logs:

```
# iasparse -f:"C:\logs\iaslog.txt"
```

4.4 Access Point Configuration for 802.1x based authentication

4.4.1 Cisco Aironet 1200 series

A Cisco Aironet AP1231G-A-K9, running firmware version 12.3(8)JEA, was used in the production of this guide.

Ensure the access point is updated with the latest firmware available from the Cisco website. Note also that, while the below instructions are specific to this series of Cisco router, they should be adaptable for use on any other 802.1x supporting router with similar features (i.e. the ability to connect to authenticate via RADIUS servers).

The AP can be configured either via SSH, console cable or configuration web page. In this case, we used the web page to configure the AP:

```
http://<IP_of_Access_Point>/
```

4.4.1.1 SSID Settings

From the AP's web page, create a new SSID in the *Security→SSID Manager*. Set the SSID to something memorable, and ensure the correct Interface is enabled (normally Radio0-802.11G).

In the *Client Authentication Settings* part, tick "Open Authentication" and choose "with EAP" in the drop down list. Also tick the "Network EAP" option, leaving the drop down menu set to "NO ADDITION". In the *Server Priorities* section further down the page, choose "Use Defaults" for the "EAP Authentication Servers".

Finally, in the *Client Authentication Key Management* section, choose Mandatory key management, checking the WPA check box. Leave the "WPA Pre-shared Key" field blank.

Scroll down to the bottom of the form, and Apply these settings (note you will need to click the top-most apply button, and **not** the one below related to the "Guest Mode/Infrastructure SSID Settings").

[4] Located on your Windows 2000/2003 install CD in the Support/Tools folder.

4.4.1.2 Encryption Settings

From the AP's web page, go to *Security→Encryption Manager*. Set the Cipher mode to one of the following:

- **TKIP**: for use with a phone set to use WPA-802.1x security type
- **AES**: for use with a phone set to use WPA2-802.1x security type

Remember the choice of setting as you will need it when configuring the phone.

4.4.1.3 RADIUS Settings

From the AP's web page, go to *Security→Server Manager*. In the "Corporate Servers" section, type in the IP address of the RADIUS server you are using. Enter in the shared secret that you specified during the setup stages of the RADIUS server. Set the Authentication and Accounting ports to 1812 and 1813 respectively. Hit the Apply button for this section.

Finally, select the Priority 1 EAP Authentication server to be the IP address of the server you just entered above. Everything else should be NONE.

4.4.2 Meru Access Points

A detailed configuration guide related to the Meru series of Wireless Access Points is available at the following link:

> *Application Notes for Meru Networks Wireless LAN System with Avaya Communication Manager and Avaya 3631 Wireless IP Telephone in a Converged VoIP and Data Network*
>
> https://devconnect.avaya.com/public/download/dyn/Meru-3631.pdf

5 Bulk Configuration

For bulk wireless security configuration of the Avaya 3631 phone, two options exist: configuration via an open access point, or via USB cable. The open access point option offers the fastest method for configuration of a set of 3631 phones, however it means a wireless access point must be setup at least temporarily to allow unauthenticated, unencrypted wireless connections (something which may not be permitted at your site). The other alternative – via USB – requires more manual setup but means no (potentially insecure) access point is required.

Configuration via USB or AP allows for the uploading of settings files and certificates (used by some of the wireless security schemes). The names of these files should be set as follows:

- **46xxsettings.txt**: the settings files for use
- **cacertx.pem**: The trusted root Certificate Authority list where *x* corresponds to the access profile that the CA certificate corresponds to (1 to 3).
- **private_key.pem**: The user certificate private key file
- **private_key_passwd.txt**: The user private key file's encryption password as plain text
- **user_cert.pem**: The user's certificate file

5.1.1 Via USB

To upload via USB, go to *Advanced Settings→Service→Backup & Restore→Download settings files*. When prompted, plug in the phone to the computer via supplied USB cable. The phone should now appear as a new USB storage device on the computer and the 5 files mentioned above can be copied into the phone's "drive". Once completed, select *Done* on the phone.

6 Appendix

6.1 OpenSSL Certificate Creation

To use these setup steps, you will need OpenSSL installed (version 0.9.8 or above). While versions of OpenSSL are available for Windows and Linux, the scripts below are designed to run under Linux.

Place the files below (in sections 6.1.1-6.1.5) into one directory and make them executable (`chmod a+x *`). Edit the `CA_CONFIG` file as appropriate for your own setup Then run:

```
# ./CA.root
```

When prompted, fill in the field information as relevant for your site. Set the "Common Name" to the name of your company. This will generate your trusted root Certificate Authority list `root.pem`. Now, depending on your configuration, you may need to run:

```
# echo 110001 > ${OUTPUTDIR}/serial
```

The value of `OUTPUTDIR` was set in the `CA_CONFIG` file. This generates a serial number for use in the next few steps. The actual value can be set to any even-length number. Now execute:

```
# ./CA.svr <server_name>
```

Once again, fill in the prompted fields as appropriate for your site. Set the "Common Name" to be your server name and give a password when prompted for a "challenge password". This will generate your server's certificate and private key in one file `<server_name>.pem`. Finally generate the client certificate(s) by running the following command for each client:

```
# ./CA.clt <client_name>
```

Supply valid fields as appropriate (the "Common Name" should be set to the client's username and is used as the Identity in TLS mode), and enter another "challenge password". The client certificate `<client_name>.pem` will then be generated

You will now have the following files created in your current directory ready for use with your RADIUS server (e.g. FreeRADIUS):

- **`root.pem`** – the trusted root Certificate Authority list (should be renamed when uploaded to the phone as described in the WPA-802.1x/WPA2-802.1x Setup Guide).
- **`<server_name>.pem`** – the server private key and certificate combined into one file
- **`<client_name>.p12`** – the client certificate and private key wrapped into one file

You will now need to execute the following commands to extract separate private key and client certificates from `<client_name>.p12`:

```
# openssl pkcs12 -in <client_name>.p12 -clcerts -nokeys –out
user_cert.pem
# openssl pkcs12 -in <client_name>.p12 -nocerts -out
private_key.pem
```

Enter a private key password when prompted. This will complete with the creation of the two files:

- **`user_cert.pem`** – the user's certificate file
- **`private_key.pem`** – the user's private key file (encrypted using the password given)

6.1.1 File: CA_CONFIG

```
# change this to the password that will be used to encrypt your private key
      files
PASSWORD=avayapwd
# change this to the location of your SSL install - CA.pl should be present
      there
PATH=/usr/lib/ssl/misc:${PATH}
# output directory as per that specified in your openssl.cnf file (under
      [ CA_default ], "dir" entry
OUTPUTDIR=./demoCA
```

6.1.2 File: CA.root

```
#!/bin/bash
```

```
# needed if you need to start from scratch otherwise the CA.pl -newca command
      doesn't copy the new
# private key into the CA directories
source CA_CONFIG
rm -rf $OUTPUTDIR
echo
      "**************************************************************************
      ********"
echo "Creating self-signed private key and certificate"
echo "When prompted override the default value for the Common Name field"
echo
      "**************************************************************************
      ********"
echo
# Generate a new self-signed certificate.
# After invocation, newreq.pem will contain a private key and certificate
# newreq.pem will be used in the next step
openssl req -new -x509 -keyout newreq.pem -out newreq.pem -passin pass:$PASSWORD
      -passout pass:$PASSWORD
echo
      "**************************************************************************
      ********"
echo "Creating a new CA hierarchy (used later by the "ca" command) with the
      certificate"
echo "and private key created in the last step"
echo
      "**************************************************************************
      ********"
echo
echo "newreq.pem" | CA.pl -newca >/dev/null
echo
      "**************************************************************************
      ********"
echo "Creating ROOT CA"
echo
      "**************************************************************************
      ********"
echo
# Create a PKCS#12 file, using the previously created CA certificate/key
# The certificate in ${OUTPUTDIR}/cacert.pem is the same as in newreq.pem.
      Instead of
# using "-in ${OUTPUTDIR}/cacert.pem" we could have used "-in newreq.pem" and
      then omitted
# the "-inkey newreq.pem" because newreq.pem contains both the private key and
      certificate
openssl pkcs12 -export -in ${OUTPUTDIR}/cacert.pem -inkey newreq.pem -out
      root.p12 -cacerts -passin pass:$PASSWORD -passout pass:$PASSWORD
# parse the PKCS#12 file just created and produce a PEM format certificate and
      key in root.pem
openssl pkcs12 -in root.p12 -out root.pem -passin pass:$PASSWORD -passout
      pass:$PASSWORD
# Convert root certificate from PEM format to DER format
openssl x509 -inform PEM -outform DER -in root.pem -out root.der
#Clean Up
rm -rf newreq.pem
```

6.1.3 File: CA.svr

```
#!/bin/bash
source CA_CONFIG
echo
      "**************************************************************************
      ********"
echo "Creating server private key and certificate"
echo "When prompted enter the server name in the Common Name field."
echo
      "**************************************************************************
      ********"
echo
# Request a new PKCS#10 certificate.
# First, newreq.pem will be overwritten with the new certificate request
openssl req -new -keyout newreq.pem -out newreq.pem -passin pass:$PASSWORD -
      passout pass:$PASSWORD
# Sign the certificate request. The policy is defined in the openssl.cnf file.
# The request generated in the previous step is specified with the -infiles
      option and
```

```
# the output is in newcert.pem
# The -extensions option is necessary to add the OID for the extended key for
      server authentication
openssl ca -policy policy_anything -out newcert.pem -passin pass:$PASSWORD -key
      $PASSWORD -extensions xpserver_ext -extfile xpextensions -infiles
      newreq.pem
# Create a PKCS#12 file from the new certificate and its private key found in
      newreq.pem
# and place in file specified on the command line
openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -out $1.p12 -clcerts -
      passin pass:$PASSWORD -passout pass:$PASSWORD
# parse the PKCS#12 file just created and produce a PEM format certificate and
      key in certsrv.pem
openssl pkcs12 -in $1.p12 -out $1.pem -passin pass:$PASSWORD -passout
      pass:$PASSWORD
# Convert certificate from PEM format to DER format
openssl x509 -inform PEM -outform DER -in $1.pem -out $1.der
# Clean Up
rm -rf newert.pem newreq.pem
```

6.1.4 File: CA.clt

```
#!/bin/bash
source CA_CONFIG
echo
      "**********************************************************************
      ********"
echo "Creating client private key and certificate"
echo "When prompted enter the client name in the Common Name field. This is the
      same"
echo " used as the Username in FreeRADIUS"
echo
      "**********************************************************************
      ********"
echo
# Request a new PKCS#10 certificate.
# First, newreq.pem will be overwritten with the new certificate request
openssl req -new -keyout newreq.pem -out newreq.pem -passin pass:$PASSWORD -
      passout pass:$PASSWORD
# Sign the certificate request. The policy is defined in the openssl.cnf file.
# The request generated in the previous step is specified with the -infiles
      option and
# the output is in newcert.pem
# The -extensions option is necessary to add the OID for the extended key for
      client authentication
openssl ca -policy policy_anything -out newcert.pem -passin pass:$PASSWORD -key
      $PASSWORD -extensions xpclient_ext -extfile xpextensions -infiles
      newreq.pem
# Create a PKCS#12 file from the new certificate and its private key found in
      newreq.pem
# and place in file specified on the command line
openssl pkcs12 -export -in newcert.pem -inkey newreq.pem -out $1.p12 -clcerts -
      passin pass:$PASSWORD -passout pass:$PASSWORD
# parse the PKCS#12 file just created and produce a PEM format certificate and
      key in certclt.pem
openssl pkcs12 -in $1.p12 -out $1.pem -passin pass:$PASSWORD -passout
      pass:$PASSWORD
# Convert certificate from PEM format to DER format
openssl x509 -inform PEM -outform DER -in $1.pem -out $1.der
# clean up
rm -rf newcert newreq.pem
```

6.1.5 File: xpextensions

```
[ xpclient_ext]
extendedKeyUsage = 1.3.6.1.5.5.7.3.2
[ xpserver_ext ]
extendedKeyUsage = 1.3.6.1.5.5.7.3.1
```

6.2 Microsoft IIS for IAS Configuration

In order to perform many of the certificate generation and manipulation functions which are required in supporting some 802.1x security modes, some configuration of the Microsoft Internet Authentication Service (IAS) server will be required. These operations are generally performed via an online webpage that the IAS server hosts on the integrated Microsoft Internet Information Service (IIS). Together, the

IAS and IIS services run on the same machine to support these operations. The configuration of IIS to support IAS is discussed below.

First, log into the Windows 2000 or 2003 IAS server and, if not already installed, install *IIS*, *Certificate Services* and *Update Root Certificates* on the Windows server (via *Control Panel→Add/Remove Windows Components*). When you install *Certificate Services*, you may be prompted to create a root certificate. Ensure you define this new certificate to be a root certificate capable of being used to create other certificates.

Now go to *Administrative Tools→Services*. Ensure that the *Internet Authentication Service* and *IIS Admin Service* are started and set to start "Automatically".

It is now necessary to enable a number of IIS features such that it can integrate properly with IAS configuration. Once again, on the IAS server go to *Administrative Tools→IIS Manager*. Expand out the tree from *IIS→local computer→Web Sites*. Right click on the "Default Web Site" and select properties. Record the *TCP port* number – you will need it later. Then right-click again on the "Default Web Site" and select "Start". The Certificate Authority web page requires a few Web Service Extensions to be running. Still in *IIS Manager*, expand to *IIS→local computer→Web Service Extensions*. Right-click on and "allow" Active Server Pages, ASP.NET, Internet Data Connector and Server Side Includes.

Confirm IIS is correctly setup to support IAS: open Microsoft Internet Explorer and navigate to:

```
http://<publicIP>:<port>/CertSrv/
```

Replace *publicIP* with the public IP address of your Windows IAS server and replace *port* with the value recorded above (as specified in IIS Manager).

6.3 Using OpenSSL to convert certificates

The Avaya 3631 only supports certificate files in PEM format. Often, an organisation may have existing certificates in another format, for example in a DER type format. Files in such a format may end with a file extension such as .crt, .cer or .der.

OpenSSL can be used in order to convert from these DER formats to PEM (as the phone requires). To do this, you will need OpenSSL installed (versions of OpenSSL are available for Windows and Linux). Then enter the following command:

```
# openssl x509 –inform der –in <cert>.cer –out <cert>.pem
```