WiMAX™ Public Key Infrastructure (PKI) Users Overview

1 Introduction

1.1 Introduction

The WiMAX™ CAs (Certificate Authorities) provide hosting of the WiMAX™ PKI (Public Key Infrastructure) hierarchy and supplies device and server certificates for use in WiMAX™ networks.

This document describes the use of these certificates by WiMAX™ devices and the process for obtaining those certificates.

1.2 X.509 Certificates

X.509 certificates and their associated keys are the documents used in a PKI system to identify and authenticate the identity of devices (SSs) and servers (AAA servers). A PKI relies on public key cryptography to digitally sign certificates by other certificates. These form a hierarchy of certificates, each signed by a higher certificate, back to a root certificate that signs itself.

The format and use of X.509 certificates are described in IETF RFC3280.

The cryptographic algorithms, such as RSA and other related specifications are in the PKCS#1 through PKCS#13 specifications, available from RSA labs.

WiMAX technology has two classes of PKI hierarchy, the device hierarchy that identifies devices and the server hierarchy that identifies AAA (Authentication, Authorization and Accounting) servers.

As the name suggests, a PKI hierarchy is arranged as a hierarchy. At the root of the hierarchy are the root signing certificates. Those roots sign subordinate CA certificates and those in turn sign either device certificates, server certificates or lower subordinate CA certificates. See Figure 1 for and example of two small PKI hierarchies.

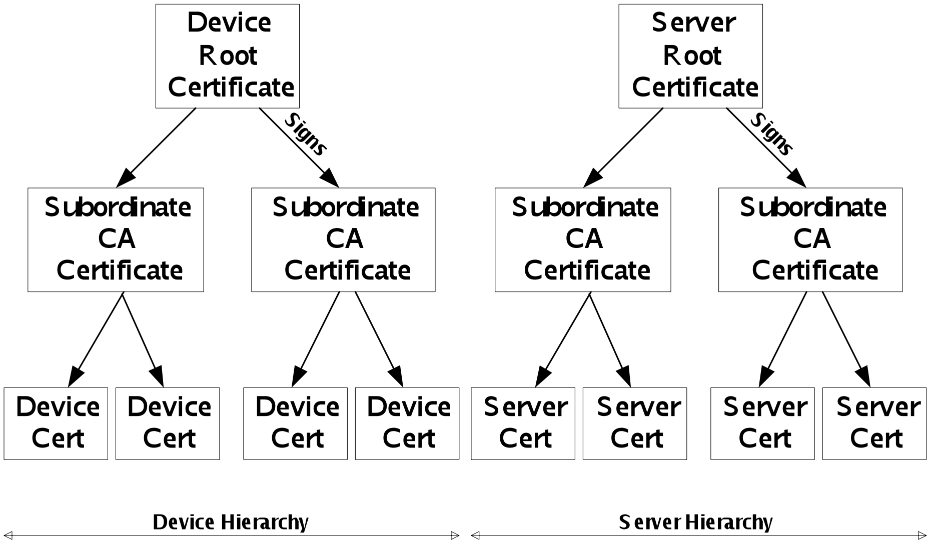

Figure 1 WiMAX PKI Hierarchies

The relationship between a signing certificate and a signed certificate is encoded in the certificates such that a computer can verify the relationship cryptographically. The issuer (the signer) has an identity that is included in the signed certificate as the "issuer identity" that matches the "subject identity" of the signing certificate. The signed certificate contains a cryptographic signature that is generated from the private key of the signer, but can be verified with the public key of the signer.

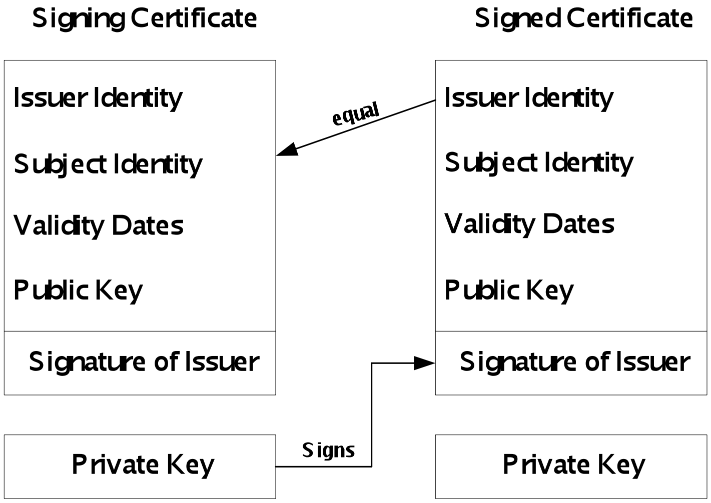

Figure 2 Relationship between a Signing and Signed Certificate

1.2.1 The Certificated Authentication Exchange

When a WiMAX Subscriber Station (SS) attaches to a WiMAX Base Station (BS) and TLS or TTLS authentication is being used, an exchange of certificates and other authentication information takes place between the SS and AAA server. The SS sends its Certificate Chain, i.e., its own certificate, the signing certificate and all the higher signing certificates back to the root certificate. Typically this will be 3 or 4 certificates; they all come from the device hierarchy. The AAA sends its certificate chain, from the server hierarchy. Typically this will be 3 certificates.

Other information is exchanged so that the SS and AAA server can prove possession of the private key associated with their certificates.

2 Provisioning Certificates in Devices and Servers

2.1 The Certificates to be provisioned in SSs and Servers

Each device must be provisioned with its certificate chain and the private key associated with its own device certificate. Also, to verify the authenticity of the server chain it receives from the network, it must have a complete list of the server root certificates that it may encounter.

Similarly, the AAA server must be provisioned with its own certificate chain and the private key associated with its own server certificate. It must also be provisioned with a complete list of the device root certificates that it may encounter in attaching devices.

The following public root certificates exist for WiMAX Public Key Infrastructure (PKI). NOTE: All of the public server root certificates MUST be installed in devices and all of the public device root certificates MUST be installed in AAA servers to guarantee future compatibility. These files are available on the WiMAX Forum website at http://www.wimaxforum.org/pkiroots/public_certificates

WiMAX Device Root	Device root created for Intel IT Flex	WiMAX Device Root.der

		WiMAX Device Root.pem
Device Root CA1	Device root created for VeriSign	Device Root CA1.der Device Root CA1.pem
Device Root CA2	Device root created for Motorola	Device Root CA2.der Device Root CA2.pem
WiMAX Server Root	Server root created for Intel IT Flex	WiMAX Server Root.der WiMAX Server Root.pem
Server Root CA1	Server root created for VeriSign (SHA1)	Server Root CA1.der Server Root CA1.pem
Server Root CA2	Server root created for VeriSign (SHA256)	Server Root CA2.der Server Root CA2.pem
Server Root CA3	Server root created for VeriSign (SHA256)	Server Root CA3.der Server Root CA3.pem

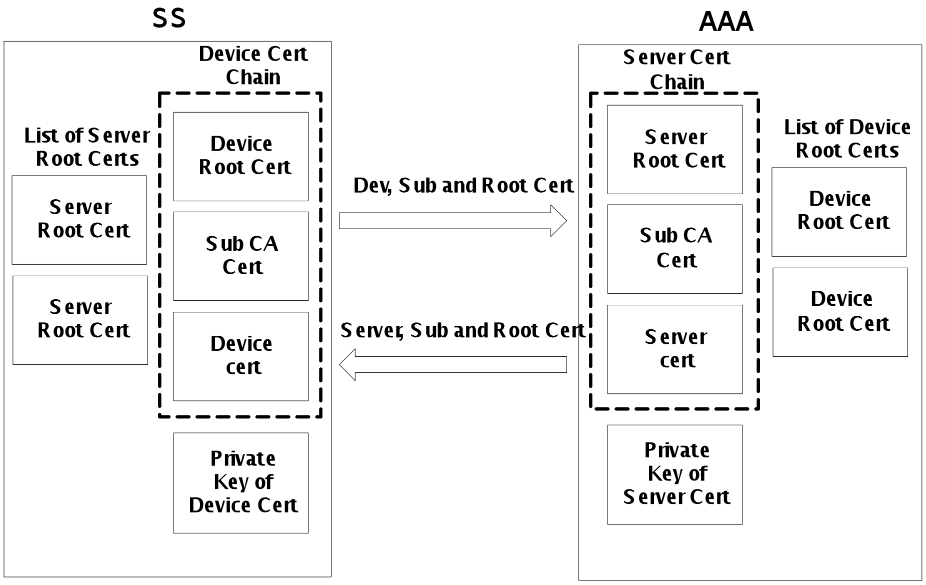

Figure 3 Certificates Provisioned in Devices

Future expansion in the list of server root certificates is unlikely to occur due to the nature of these certificates. The list of device root certificates may expand in the future and AAA servers will need to be provisioned with these as they become available.

2.1.1 Certificates and Keys to Obtain from a CA and WiMAX

When obtaining a device certificate from a CA to provision in a device, the CA must supply:

The root certificate in the device's certificate chain

The first subordinate CA certificate signed by the root.

(If present) the second subordinate signing certificate signed by the first subordinate CA certificate.

The device certificate itself, signed by the first or second subordinate CA certificate

The device certificate private key.

The list of server root certificates to provision in the device should be obtained from the WiMAX Forum..

When obtaining a server certificate from a CA, to provision in an AAA server, the CA must supply:

The root certificate in the server's certificate chain

The first subordinate CA certificate signed by the root.

(If present) the second subordinate signing certificate signed by the first subordinate CA certificate.

The server certificate itself, signed by the first or second subordinate CA certificate

The private key of the server certificate

The list of device root certificates to provision in the server should be obtained from the WiMAX Forum.

The server and device chain subordinate certificate(s) may change frequently, E.G. to limit the size of CRLs (certificate revocation lists). Therefore a fresh copy of the subordinate certificates should be obtained with each certificate delivery to ensure that consistent certificate chains are provisioned.

3 Obtaining Certificates

The process for ordering WiMAX PKI certificates has transitioned as of May, 2009 as the WiMAX Forum is no longer in the business of processing certificate orders. Motorola and VeriSign are now providing WiMAX PKI certificate services. To learn about the process for obtaining certificates from Motorola or VeriSign please visit the WiMAX PKI webpage. Members of the WiMAX forum can log in and follow supplier links. If you are not a member of the WiMAX forum and wish to learn more about the benefits of membership or wish to become a member please visit the WiMAX Forum Membership webpage.

Setting up an Authorized User

An "Authorized User" is a Device Manufacturer or a Network Operator that the WiMAX Forum, as authorization administrator under authority of the WiMAX PA, has authorized to receive PKCs from a CA. Authorized Users must be set up before the recipient can request device or server certificates.

The process for setting up Authorized Users and ordering certificates is available at the WiMAX Forum website for users who are employees of active member companies of the WiMAX Forum.

The WiMAX Forum has established a Public Key Infrastructure ("PKI"), the WiMAX™ PKI, which is described in the Governing Documents and is operated under the supervision of the WiMAX Policy Authority. The WiMAX PKI provides a mechanism that permits Authorized Users to obtain PKCs through Certificate Authorities. In support of maintaining a high level of security and integrity for the WiMAX PKI and the networks established and operated using the PKCs, the WiMAX PA requires that each applicant wishing to receive PKCs execute a WiMAX PKI License Agreement before the applicant is approved as an Authorized User. Licensee wishes to become an Authorized User and the WiMAX Forum, as administrator, is willing to approve Licensee as an Authorized User subject Licensee's faithful performance of all of its obligations described in the License Agreement.

3.1 Requesting Device Certificates

Once an Authorized User has been approved as an authorized user, orders for device certificates can be requested at the WiMAX Forum website. Authorized Users log in to the members area and can contact Motorola or VeriSign via a web link and will be directed by each company in accordance with their procedures.

3.2 *Requesting a Server Certificate*

Once an Authorized User has been approved, orders for server certificates can be requested at the WiMAX Forum website. Authorized Users log in to the members area and can contact VeriSign via a web link and will be directed in accordance with their procedures.

3.3 *Retrieving root certificate lists*

The WiMAX Forum publishes the approved WiMAX root certificates on its web site.

4 Operating PGP and GnuPG

4.1 *Overview of the use of PGP and GnuPG*

PGP is a commercial email and file encryption and signing product. It makes use of PGP public/private key pairs. PGP is used to protect communication between certificate Authorized Users and the WiMAX CA. PGP is available on multiple platforms. It is commonly used on Windows systems and it is in this context that it is described in this document.

GnuPG is a free toolset compatible with PGP. It is available on multiple platforms. It is typically distributed with Linux and Unix distributions and it is in this context that it is described in this document.

An Authorized User must establish a PGP key pair to protect communications with the WiMAX CA. This may be either a key pair for this specific purpose or may be a key pair normally used by the Authorized User for email communications.

The public part of the Authorized User's key pair is supplied to the WiMAX CA during the setting up of an Authorized User, as per section 3.1.

4.2 *Obtaining GnuPG*

GPG in source form is available from the GnuPG website http://www.gnupg.org/. At the time of writing the most recent primary version is 1.4.8. The most recent 2.0 series version is 2.0.8. GPG is available in binary package form for many distributions. On Fedora and RedHat it is available through the Applications->Add/Remove Software menu item. Search for gnupg.

On systems with Yum, from the command line as root, type:

```
#> yum install gnupg
```

4.3 *Obtaining PGP*

PGP is available from PGP Corporation. It can be purchased from their website at http://www.pgp.com. The "Windows Desktop Professional" product is the one used by the WiMAX CA. Other PGP products should be compatible.

At the time of writing, "Windows PGP Desktop Professional" product could be purchased from http://www.pgp.com by following the "Purchase" link, then the "PGP Online Store" link, then the "PGP Desktop Professional" link then selecting the preferred license type and proceeding with the ordering process.

PGP is also available for Macintosh OSX.   .

4.4 *Creating a key pair in GPG*

A signing key pair in GPG is made using the `gpg --gen-key` command. This will begin an interactive session to generate the key. This should be followed with the `gpg --edit-key <YourName>` command followed by addkey and setpref S9 H8 to add and configure an encryption key.

The following example session show what to type in bold, comments are in italics.

```
[operator1@dj-desk1 ~]$ gpg --gen-key

gpg (GnuPG) 1.4.7; Copyright (C) 2006 Free Software Foundation, Inc.

This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions.  See the file COPYING for details.
gpg: directory “/home/operator1/.gnupg“ created
gpg: new configuration file “/home/operator1/.gnupg/gpg.conf” created
gpg: WARNING: options in “/home/operator1/.gnupg/gpg.conf” are not yet active during this run
gpg: keyring “/home/operator1/.gnupg/secring.gpg” created
gpg: keyring “/home/operator1/.gnupg/pubring.gpg” created
Please select what kind of key you want:
   (1) DSA and Elgamal (default)
   (2) DSA (sign only)
   (5) RSA (sign only)
Your selection? 5              [This selects an RSA key]
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)              [Hit “enter” here]
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0)                         [Hit  “enter” here]
Key does not expire at all
Is this correct? (y/N) y                      [Confirm here]

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

[Enter the same name and email that will go on the Authorized User form]
Real name: YourName
Email address: YourEmailName@YourEmailAddress.com
Comment:
You selected this USER-ID:
    "YourName <YourEmailName@YourEmailAddress.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

Enter passphrase: [Enter you passphrase here, keep it secure]
```

```
Repeat passphrase: [Enter you passphrase here, keep it secure]

We need to generate a lot of random bytes.  It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++   [Type randomly, click on the screen etc.]
+++++
gpg: /home/operator1/.gnupg/trustdb.gpg: trustdb created
gpg: key DFEFCCF1 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   2048R/DFEFCCF1 2008-01-11
      Key fingerprint = 50F5 E7FA 9080 8030 91C0  66BC 9D07 D134 DFEF CCF1
uid                  YourName <YourEmailName@YourEmailAddress.com>

Note that this key cannot be used for encryption.  You may want to use
the command "--edit-key" to generate a subkey for this purpose.
[operator1@dj-desk1 ~]$
```

```
[operator1@dj-desk1 ~]$ gpg --edit-key YourName
gpg (GnuPG) 1.4.7; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions.  See the file COPYING for details.

Secret key is available.

pub  2048R/DFEFCCF1  created: 2008-01-11  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
[ultimate] (1). YourName <YourEmailName@YourEmailAddress.com>

Command> addkey           [This is to add an encryption key]
Key is protected.

You need a passphrase to unlock the secret key for
```

```
user: "YourName <YourEmailName@YourEmailAddress.com>"
2048-bit RSA key, ID DFEFCCF1, created 2008-01-11

Enter passphrase: [Enter your passphrase here]
user: "YourName <YourEmailName@YourEmailAddress.com>"
2048-bit RSA key, ID DFEFCCF1, created 2008-01-11

Please select what kind of key you want:
   (2) DSA (sign only)
   (4) Elgamal (encrypt only)
   (5) RSA (sign only)
   (6) RSA (encrypt only)
Your selection? 6              [This selects an RSA key]

RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) [Hit “enter” here]
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y
Really create? (y/N) y
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++
.....+++++

pub  2048R/DFEFCCF1  created: 2008-01-11  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  2048R/7FA3CDFD  created: 2008-01-11  expires: never       usage: E
[ultimate] (1). YourName <YourEmailName@YourEmailAddress.com>

Command>setpref S9 H8  [This is to set the encryption and signing algorithms to AES and SHA-256]
```

```
Set preference list to:
   Cipher: AES256, 3DES
   Digest: SHA256, SHA1
   Compression: ZIP, Uncompressed
   Features: MDC, Keyserver no-modify
Really update the preferences? (y/N) y

You need a passphrase to unlock the secret key for
user: "YourName <YourEmailName@YourEmailAddress.com>"
2048-bit RSA key, ID DFEFCCF1, created 2008-01-11
Enter passphrase: [Enter your passphrase here]

pub  2048R/DFEFCCF1  created: 2008-01-11  expires: never       usage: SC
                     trust: ultimate      validity: ultimate
sub  2048R/7FA3CDFD  created: 2008-01-11  expires: never       usage: E
[ultimate] (1). YourName <YourEmailName@YourEmailAddress.com>

Command> quit
Save changes? (y/N) y
[operator1@dj-desk1 ~]$
```

4.5 Creating a key Pair in PGP

The PGP installation process guides the installer through the process of initial key creation. When in PGP, select File-> New PGP Key…

Click on “Next>” at the introduction.

Enter the name and primary email in the “Name and Email Assignment” screen. This name and email must match the name field and email address that is entered in the “New Authorized User Request Form” as described in section 3.1.

The key generated will default to a 2048 bit RSA key, AES and SHA-2-256 signature. This matches the security level in the delivered certs and so it is permitted but not necessary to alter these defaults.

Click “Next>“

Enter a passphrase into the “Passphrase Assignment” screen. Keep a secure record of this passphrase. Without it, the keys will be unusable.

Click “Next>“ . The keys will be generated.

Click “Next>“ . From here you may submit the keys to the global directory if you wish.