SpeedTouch^{TM}610 Certificates - Practical Configuration

Author: Heyrman Alfons

Date: January 2003

Edition: v1.0

Abstract: The goal of this document is to explain the implementation details of the SpeedTouch^{TM}610 regarding the enrollment and configuration of certificates. Short theoretical explanations will be provided where necessary but the main focus of this document will be on the practical configuration. For more in-depth explanation and advanced setups, the user is invited to read - among others - the SpeedTouch^{TM} White Paper "Understanding the IP Security Protocol Suite" and "PKI", and the Application Note "SpeedTouch^{TM}610 IPSec Configurations Demystified".

Note that to use the IP Security and IPSec enabled VPN features of the SpeedTouch^{TM}610, the IPSec VPN software key must be installed. See the Application Notes "SpeedTouch™610 Operation and Maintenance" and "SpeedTouch^{TM}610 Software Activation Keys" or ask your Service Provider for more information.

Applicability: This application note applies to the following SpeedTouch^{TM} products:

- The SpeedTouch^{TM}610 Business ADSL/POTS Router
- The SpeedTouch^{TM}610i Business ADSL/ISDN Router
- The SpeedTouch^{TM}610s Business SHDSL Router
- The SpeedTouch^{TM}610v Business VDSL Router.

Conventions: As per convention, in this document all SpeedTouch^{TM}610 Business DSL Router variants will be referred to as SpeedTouch^{TM}610, unless a specific variant is concerned.

Updates: Due to the continuous evolution of DSL technology, existing products are regularly upgraded. For more information on the latest technological innovations, software upgrades, and documents, please visit the SpeedTouch^{TM} web site at:

http://www.speedtouch.com

1 CERTIFICATES

1.1 What are certificates?

Certificates are documents that contain data binding a public key to an end-entity or individual and are typically used to identify that person of object." This person or object is commonly referred to as end-entity (EE).

They are issued by a trusted third party, a Certificate Authority (CA), acting as a root source for establishing a chain of trust between individuals/objects also granted certificates by that CA. The certificates are tamper-proof and cannot be forged. Certificates authenticate that their holders, devices or users are truly who or what they claim to be.

In an IPsec scenario certificates are used to verify a user's or device's identity, once their identity is proven, the authentication process of the Phase 1 is successful and further tunnel setup can proceed.

1.2 Intercommunication Protocols

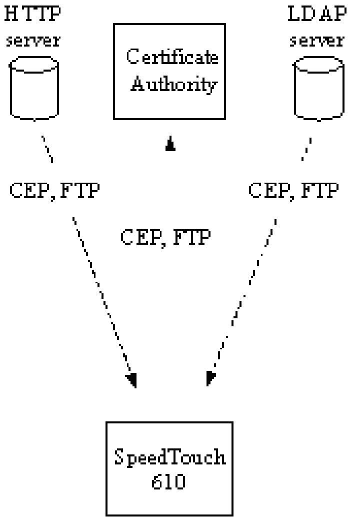


Figure 1: Intercommunication Protocols

Certificates are generated by the Certification Authority and centrally stored. The possible access protocols between the different entities are as outlined above.The CEP (Certificate Enrollment Protocol) allows for online certificate retrieval and is supported on the SpeedTouch^{TM}610.

1.3 Overview

Below is a brief overview of the certificate logic implemented on the SpeedTouch^{TM}610. As we can see, certificate importation can be done either manually (offline) or by using the Certificate Enrollment protocol (online). Both methods are supported for certificates and CRLs (Certificate Revocation Lists). The offline certification process requires the user to manually send his request to the Certification Authority, whereas for the online certification the request is automatically send using the Certificate Enrollment protocol (CEP).

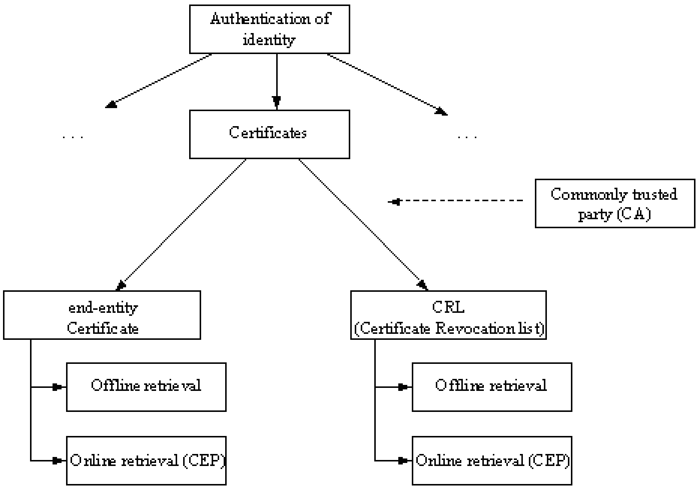


Figure 2: Overview of certificates

1.4 Chain of Trust

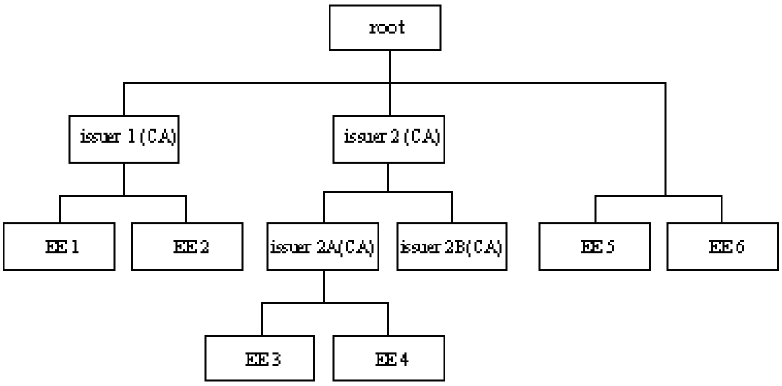


Figure 3: Chain of Trust

The concept “chain of trust” means that, despite the fact that the end-entity certificates are signed by different issuers (EE1 is signed by issuer1 and EE3 by issuer 2A), they share a common trusted party (root). When checking the certificate for validity, there is a chain of trust that travels upward until a common trusted party is found (root).

1.5 Cross-Certification (Multiple Root CA Certificates)

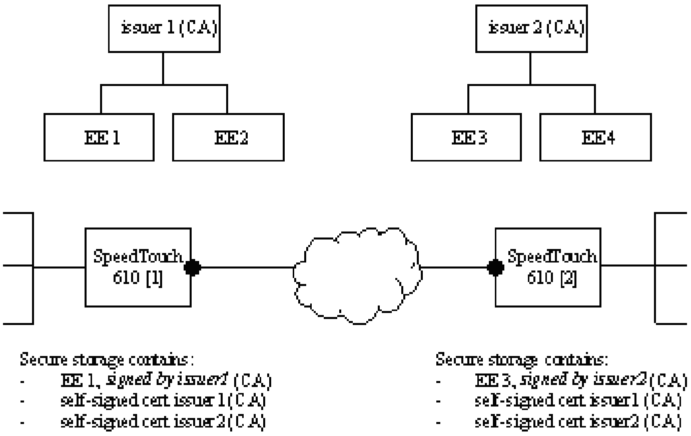

issuer 1 (CA)
issuer 2 (CA)
EE 1
EE2
EE 3
EE4
SpeedTouch 610 [1]
SpeedTouch 610 [2]
Secure storage contains:
- EE 1, signed by issuer1 (CA)
- self-signed cert issuer 1 (CA)
- self-signed cert issuer 2 (CA)
Secure storage contains:
- EE 3, signed by issuer2 (CA)
- self-signed cert issuer1 (CA)
- self-signed cert issuer2 (CA)

Figure 4: Cross-Certification

Both SpeedTouch^{TM}610 devices are certified by different issuers. The two issuer CA's in the example are not linked to each other by a chain of trust.

When using certificates as authentication method to establish a secure tunnel, one must also import the issuer's (root) certificate of the remote SpeedTouch^{TM}610. Importing this certificate means you trust that issuer and will accept certificates issued by it.

The issuers can be of the same CA type or not. The SpeedTouch^{TM}610 has been tested for interoperability with the Netscape, Entrust and VeriSign CA types.

Note A combination of chain of trust and cross-certification is also supported on the SpeedTouch^{TM}610.

2 REQUEST CERTIFICATES

2.1 Theory

Enrollment is the process of creating and submitting a certificate request to a Certificate Authority and, once granted, the Authority making available the new certificate. For interoperabilility purposes, several standards have been developed. RSA Laboratories coordinates the development and maintenance of PKCS (Public Key Cryptography Standards), a series of public key cryptography specifications produced in co-operation with system developers worldwide. Supported on the SpeedTouch™610 are:

- PKCS#10
  A certificate request syntax standard: a standard syntax for requesting certification of a public key and a distinguished name.
- PKCS#7
  A general syntax for data that may have cryptography applied to it, such as certificates.
- PKCS#12
  Defines the personal information exchange syntax standard, which specifies a portable format for storing a user's private key and certificates.

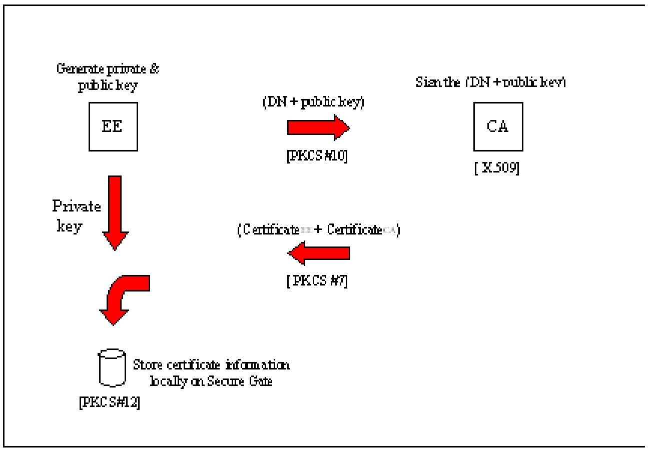

Generate private & public key
EE
Private key
Store certificate information locally on Secure Gate
[PKCS#12]
(DN + public key)
[PKCS #10]
(Certificate + Certificate)
[PKCS #7]
Sign the (DN + public key)
CA
[X.509]

Figure 5: Usage of different exchange formats

When launching a certificate request on the SpeedTouch™610 (= the end-entity or EE), a PKCS#10 Base 64 encoded text block is generated. On the CA, this PCKS#10 request is granted by the CA administrator and packed into PKCS#7 format. Once the signed certificate is imported onto the SpeedTouch™610, it will be stored in PKCS#12 formatted file.

When a PKCS#10 request arrives at the CA, following steps are done prior signing:

1. Verify whether this EE is indeed allowed to be certified.
2. Authenticate the message using the public key of the EE.
3. Sign the [public keyEE + ID] with the CA's private key (= the actual certification).

Certificates can be imported either by copy/pasting the PKCS#7 information into the CLI interface or by putting a file on the SpeedTouch™610 filesystem and importing it. The copy/paste example will be illustrated for the offline importation of the end-entity certificate whereas for the CRL importation example the file method will be used.

Note In order to use certificates, the correct time must be configured at the device. In most cases, certificate importation fails due an incorrect setting of the time or date.

2.2 Offline Certificate Importation

2.2.1 Generate key pair

```
[ipsec cert]=>request
subjectdn = cn=EndEntityCert, o=Thomson
[force] =
[ipsec cert]=>
```

A private and public key pair are generated and stored onto the SpeedTouch^{TM}610 in PKCS#12 format. As output, the DN and public key in PKCS#10 format are listed (output via CLI):

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBVzCBwQIBADAYMRYwFAYDVQQDEw1FbmRFbnRpdHlDZXJ0MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDAEyw0
qyN0NOyCcHbBxQiI1eoyHzdcF2p4UO7nOc5+765KDcox+ke3yOVsDfPt0l6xRwkoCle1iwxF4VYlX4w4vaDHBjyfmREC
+OWDX3UVNRnjJjxbMRJwvjIAbufM97PeKCvWFTdCCFkX8tKI03vhmxvN4mSbpxtKoJRRwIDAQABoAAwDQYJKoZIhvcNA
QEEBQADgYEAIUmYHqYpBNvjVYEXIliBnmwhBbveCMNZum3D9Gnsgy7A3zYusrgfKCyTIUl56SaO+tC9Q7iS5SUAcxihH
tQbRyQsdIKnWXCH0bib2yLclV0B8qCLaDHXE+nJO9Lnh1PAuZBCqL1TnC1ejLhre+iOV61hVTXcotGNm4NAsUs1s=
-----END NEW CERTIFICATE REQUEST-----
```

2.2.2 Sign Request

The request in PKCS#10 format now has to be sent to the Certification Authority (CA). Depending on the CA type used, the certificate request can be sent via e.g. HTTP, FTP, floppy disk, … Please refer to your CA manual on how to import and sign a certificate request.

Below an example of a signed certificate in PKCS#7 format is depicted:

- Base 64 encoded certificate

```
-----BEGIN CERTIFICATE-----
MIICXTCCAgegAwIBAgIBIzANBgkqhkiG9w0BAQQFADA9MQswCQYDVQQGEwJCRTEQMA4GA1UEChMHQWxjYXRlbDEc
MBoGA1UEAxMTQ2VydGlmaWNhdGUgTWFuYWdlcjAeFw0wMTA4MjIxMjIyMTVaFw0wMjA4MjIxMjIyMTVaMD8xCzAJ
BgNVBAYTAkJlMRAwDgYDVQQKEwdBbGNhdGVsMQwwCgYDVQQLEwNjcGUxEDAOBgNVBAMTB3ByZXNpdGEwgZ8wDQYJ
KoZIhvcNAQEBBQADgY0AMIGJAoGBAJQaSWLjgi+7Y0QE2t52FtA9uRR6kGGcVV0EAtECYoK5fGDP8mu5B8LdxOZQ
tOpenHXyPAhASijfLi4XzTZmjg5Zel2hene6a292k03s0oBxZelqkFD7L5prvmZ+0gdYihbnu+QPSC13xhEzUhkM
9RbMj4MuDTnYZGJch4FjDAgMBAAGjgaswgagwEQYJYIZIAYb4QgEBBAQDAgZAMA4GA1UdDwEBwQEAwIE8DAfBgNV
HSMEGDAWgBT3M85n9aLJ0fMyNpPknxc7lF7l2jBDBgNVHR8EPDA6MDigNqA0pDIwMDEQMA4GA1UEChMHQWxjYXRl
bDEcMBoGA1UEAxMTQ2VydGlmaWNhdGUgTWFuYWdlcjAdBgNVHREEFjAUgRJwcmVzaXRhQGFsY2F0ZWwuYmUwDQYJ
KoZIhvcNAQEEBQADQQASSMJfkq4WkROH2A0Jw0su1n2AL1mjITJ3+dbm2MKmGsrnPmwtDnMn20DJXOw7tbJygo+f
-----END CERTIFICATE-----
```

- Base 64 encoded certificate with CA certificate chain in pkcs7 format:

```
-----BEGIN CERTIFICATE-----
MIIEdQYJKoZIhvcNAQcCoIIEZjCCBGICAQExADAPBgkqhkiG9w0BBwGgAgQAoIIE
RjCCAeEwggGLoAMCAQICAQEwDQYJKoZIhvcNAQEFBQAwPTELMAkGA1UEBhMCQkUx
EDAOBgNVBAoTB0FsY2F0ZWwxHDAaBgNVBAMTE0NlcnRpZmljYXRlIE1hbmFnZXIw
HhcNMDEwNzI1MjIwMDAwWhcNMDMwNzI1MjIwMDAwWjA9MQswCQYDVQQGEwJCRTEQ
MA4GA1UEChMHQWxjYXRlbDEcMBoGA1UEAxMTQ2VydGlmaWNhdGUgTWFuYWdlcjBc
MA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCcZDy94kMbUCjoDi5E6WSi7w9vzaU6Z8/D
Zl4ytw4KBbfAXQe0PgNKAaRKbbXfiQ4oxjMYxGCAbZ3yF6hbKHMNAgMBAAGjdjB0
MBEGCWCGSAGG+EIBAQQEAwIABzAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBT3
M85n9aLJ0fMyNpPknxc7lF7l2jAfBgNVHSMEGDAWgBT3M85n9aLJ0fMyNpPknxc7
lF7l2jAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQEFBQADQQAcfTUQT0ukOFuI
VHlaaumKG+Pzb5VZyN+9WI2kK3fx/hXxVodXW7u8EnFgVSNwZ6SVDAAqUo4QqhkT
0bY/sHjOMIICXTCCAgegAwIBAgIBIzANBgkqhkiG9w0BAQQFADA9MQswCQYDVQQG
EwJCRTEQMA4GA1UEChMHQWxjYXRlbDEcMBoGA1UEAxMTQ2VydGlmaWNhdGUgTWFu
YWdlcjAeFw0wMTA4MjIxMjIyMTVaFw0wMjA4MjIxMjIyMTVaMD8xCzAJBgNVBAYT
AkJlMRAwDgYDVQQKEwdBbGNhdGVsMQwwCgYDVQQLEwNjcGUxEDAOBgNVBAMTB3By
ZXNpdGEwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJQaSWLHjgi+7/Y0QE2t
52FtA9uRR6kGGcVV0EAtECYoK5fGDP8mu5B8LdxOZQtOpenHXyPAhASijfLi4XzT
Zmjg5Zel2hene6a292k03s0oBxZe/lqkFD7L5prvmZ+0gdYihbnu+QPSC13xhEzU
hkM9RbMj4MuDTnYZGJch4FjDAgMBAAGjgaswgagwEQYJYIZIAYb4QgEBBAQDAgZA
MA4GA1UdDwEB/wQEAwIE8DAfBgNVHSMEGDAWgBT3M85n9aLJ0fMyNpPknxc7lF7l
2jBDBgNVHR8EPDA6MDigNqA0pDIwMDEQMA4GA1UEChMHQWxjYXRlbDEcMBoGA1UE
AxMTQ2VydGlmaWNhdGUgTWFuYWdlcjAdBgNVHREEFjAUgRJwcmVzaXRhQGFsY2F0
ZWwuYmUwDQYJKoZIhvcNAQEEBQADQQASSMJfkq4WkROH2A0Jw0su1n2AL1mjITJ3
+dbm2MKmGsrnPmwtDnMn20DJXOw7tbJygo+fUmJg/Vsa/PzpbHuRMQA=
-----END CERTIFICATE-----
```

Note When using the Netscape CA for certification, two PKCS#7 base 64 encoded text blocks are available after signing: one only containing the signed end-entity certificate and a second one containing both the CA and the end-entity's certificate.

2.2.3 Import Certificate

One must first import the CA certificate onto the SpeedTouch^{TM}610 prior to importing the end-entity certificate.

Therefore execute the command below:

```
[ipsec cert]=>import
```

And paste your certificate. End with CTRL+D (and an approval of the import).

The following shows an example of importing a certificate onto the SpeedTouch^{TM}610:

1 Copying & pasting the PKCS#7 text blob containing both the CA & EE cert to the SpeedTouch^{TM}610 CLI:

```
-----BEGIN CERTIFICATE-----
MIIEdQYJKoZIhvcNAQcCoIIEZjCCBGICAQExADAPBgkqhkiG9w0BBwGgAgQAoIIE
RjCCAeEwggGLoAMCAQICAQEwDQYJKoZIhvcNAQEFBQAwPTELMAkGA1UEBhMCQkUx
EDAOBgNVBAoTB0FsY2F0ZWwxHDAaBgNVBAMTE0NlcnRpZmljYXRlIE1hbmFnZXIw
HhcNMDEwNzI1MjIwMDAwWhcNMDMwNzI1MjIwMDAwWjA9MQswCQYDVQQGEwJCRTEQ
MA4GA1UEChMHQWxjYXRlbDEcMBoGA1UEAxMTQ2VydGlmaWNhdGUgTWFuYWdlcjBc
MA0GCSqGSIb3DQEBAQUAA0sAMEgCQQCcZDy94kMbUCjoDi5E6WSi7w9vzaU6Z8/D
Zl4ytw4KBbfAXQe0PgNKAaRKbbXfiQ4oxjMYxGCAbZ3yF6hbKHMNAgMBAAGjdjB0
MBEGCWCGSAGG+EIBAQQEAwIABzAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBT3
M85n9aLJ0fMyNpPknxc7lF7l2jAfBgNVHSMEGDAWgBT3M85n9aLJ0fMyNpPknxc7
lF7l2jAOBgNVHQ8BAf8EBAMCAYYwDQYJKoZIhvcNAQEFBQADQQAcfTUQT0ukOFuI

0bY/sHjOMIICXTCCAgegAwIBAgIBIzANBgkqhkiG9w0BAQQFADA9MQswCQYDVQQG
EwJCRTEQMA4GA1UEChMHQWxjYXRlbDEcMBoGA1UEAxMTQ2VydGlmaWNhdGUgTWFu
YWdlcjAeFw0wMTA4MjIxMjIyMTVaFw0wMjA4MjIxMjIyMTVaMD8xCzAJBgNVBAYT
AkJlMRAwDgYDVQQKEwdBbGNhdGVsMQwwCgYDVQQLEwNjcGUxEDAOBgNVBAMTB3By
ZXNpdGEwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJQaSWLHjgi+7/Y0QE2t
52FtA9uRR6kGGcVV0EAtECYoK5fGDP8mu5B8LdxOZQtOpenHXyPAhASijfLi4XzT
Zmjg5Zel2hene6a292k03s0oBxZe/lqkFD7L5prvmZ+0gdYihbnu+QPSC13xhEzU
hkM9RbMj4MuDTnYZGJch4FjDAgMBAAGjgaswgagwEQYJYIZIAYb4QgEBBAQDAgZA
MA4GA1UdDwEB/wQEAwIE8DAfBgNVHSMEGDAWgBT3M85n9aLJ0fMyNpPknxc7lF7l
2jBDBgNVHR8EPDA6MDigNqA0pDIwMDEQMA4GA1UEChMHQWxjYXRlbDEcMBoGA1UE
AxMTQ2VydGlmaWNhdGUgTWFuYWdlcjAdBgNVHREEFjAUgRJwcmVzaXRhQGFsY2F0
ZWwuYmUwDQYJKoZIhvcNAQEEBQADQQASSMJfkq4WkROH2A0Jw0su1n2AL1mjITJ3
+dbm2MKmGsrnPmwtDnMn20DJXOw7tbJygo+fUmJg/Vsa/PzpbHuRMQA=
-----END CERTIFICATE-----Read 1516 bytes from stdin.
```

2 Pressing CTRL-D and approving the import:

```
Object #1: Certificate
  Subject: cn=Certificate Manager, o=Thomson, c=BE
  Issuer: cn=Certificate Manager, o=Thomson, c=BE
  notBefore: Wed Jul 25 22:00:00 2001  notAfter: Fri Jul 25 22:00:00 2003
Object #2: Certificate
  Subject: cn=EndEntityCert, o=Thomson
  Issuer: cn=Certificate Manager, o=Thomson, c=BE
  notBefore: Fri Jul 27 12:45:49 2001  notAfter: Sat Jul 27 12:45:49 2002

Are you sure you want to import this certificate/CRL ? (y/n) :y
[ipsec cert]=>
```

2.2.4 Was the import successful?

Verify with the commands shown below whether the certificate was successfully imported in the secure storage. Looking at the details of the certificate, the validity can be checked.

```
[ipsec cert]=>list
Item  Distinguished Name  Type            Issuer      Serial Number

1    cn=EndEntityCert, o=Thomson         CERT  Cert #2     113
2    cn=Certificate Manager, o=Thomson, c=BE   CERT        Self-signed 1

[ipsec cert]=>
[ipsec cert]=>list item=1
        Status:  Valid
       Version:  V3
        Issuer:  cn=Certificate Manager, o=Thomson, c=BE
     Serial No:  113
 Validity Date:  not before: Fri Jul 27 12:45:49 2001
                 not after:  Sat Jul 27 12:45:49

       Subject:  cn=EndEntityCert, o=Thomson
    Extensions:  _keyUsage: digitalSignature, keyEncipherment,
                 _authorityKeyIdentifier
                 _CRLDistributionPoints
             1. ldap://192.6.11.50:389/cn=Certificate Manager, o=Thomson
[ipsec cert]=>
```

2.3 Online Certificate Importation

2.3.1 Configure CEP Parameters

A minimum of configuration settings (Enrollment URL and SubjectDN) must be configured before CEP enrollment can be initiated. An error message will be displayed when the minimum information is not supplied.

The CEP options listed below are CA dependant and should be modified based on the user's PKI environment.

```
[ipsec cert cep]=>config
Enrollment URL     : http://138.203.14.183/cgi-bin/pkiclient.exe
CA Identity string :
CA MD5 Fingerprint :
HTTP proxy         :
Subject DN         : cn=EndEntityCert, o=Thomson
Key length         :
Challenge Password :
X509v3 extension
  Email Address    :
  DnsName          :
  Ip Address       :
  Alt Subject DN   :
CheckNonce         : yes
CheckTransactionID : yes
KeyUsage extension :
[ipsec cert cep]=>
```

Below a description is given of the two mandatory parameters, i.e. the “Enrollment URL” and the “Subject DN”.

- [url = <quoted string>]
  Is the CEP enrollment URL which represents the URL pointing to the CEP script on the CA server.
  It’s usually in the form “http://<host>[:<port>]/cgi-bin/pkiclient.exe”.
  This entry must have a value, e.g.: “EnrollmentURL=http://192.6.11.183/cgi-bin/pkiclient.exe”.
  - [address] is a numeric address, do not enter a DNS name
  - [port] is the port number (optional port 80 is assumed)
  - [path] is the path to the CA’s CEP script
- [subjectdn = <Distinguished Name (RFC1779): e.g. cn=xxx, c=be>]
  Is the distinguished name (RFC1779) for the certificate. This reflects the subject name for the requested certificate. The value of this entry must be a valid distinguished name in string representation (RFC 1779).
  The SubjectDN can include common name (cn=), organization unit (ou=), organization name (o=), locality (l=), province or state (st=), and country (c=).
  Use commas to separate the items, and enclose all items in quotation marks.

2.3.2 Generate key pair & launch CEP request

The command below will generate the private and public keys and send the Distinguished Name and public key to the CA for signing:

```
=>:ipsec cert cep request
=>
```

2.3.3 Check fingerprint

When the CEP request is launched, we have to wait for the Certification Authority to sign it. When checking for the granted certificate, we will use the fingerprint to identify our request.

```
:ipsec cert cep list
      Request info :
      Dn : cn=EndEntityCert, o=Thomson
      Fingerprint : 7A:DE:85:3C:B8:3C:84:36:AF:89:3C:FA:59:F1:12:37
```

Note Speak to your CA administrator for the correct Finger print value of your CA to verify against.

2.3.4 Import signed certificate

After the certificate is signed, two means exist to import it onto the device: either manually or automatically:

- Manual
  Contact the CA and check the certificate status> If it granted, retrieve and imports the certificate onto the SpeedTouch^{TM}610:

  ```
  :ipsec cert cep resubmit
  ```

- Automatic
  Every 8 minutes, an automatic check is done verifying the status of the certificate. If it is granted, the certificate is retrieved and imported.

Note You can cancel the pending CEP request using the below command:

```
:ipsec cert cep cancel
```

2.3.5 Importation successful?

Use the same commands as specified for the offline enrollment:

```
:ipsec cert list
```

or (with <x> being the index number of the certificate):

```
:ipsec cert list item <x>
```

3 REQUEST CERTIFICATE REVOCATION LIST

3.1 Theory

Certificates have only limited validity. After the certificate's validity period is expired, it cannot be used anymore for tunnel setup. Another way for a certificate to become invalid is if it is listed in the CRL (Certificate Revocation List). When a certificate gets compromised, the Certification Authority will revoke it and it will become part of the CRL. For example, when an employee leaves an organization, his certificate will be revoked by the administrator and the CA will add it to the CRL. After revocation, the former employee will not be able anymore to call in onto the company network. Each IPSec device will regularly update its list of revoked certificates by contacting the CA.

During SA negotiation if the certificate of the remote IPSec peer is listed in the CRL, the authentication process will fail and secure tunnel setup will be refused.

There are two methods for getting an updated CRL onto the device; Offline and Online.

3.2 Offline CRL retrieval

3.2.1 Generate the CRL

On the CA, generate the CRL and put it on of the following file formats:

- base64 encoded
- binary data
- ASN1 encoded text data.

3.2.2 Transfer to filesystem

The file containing the CRL information now needs to be transferred to the SpeedTouchTM610's file system. Execute the below steps:

- go to the local directory on the machine where the CRL file is located
- ftp <IP address SpeedTouchTM610>
- cd /dl
- put <name CRL file>.

3.2.3 Import the CRL

Execute the import command specifying the CRL's filename in the Cli command.

```
:ipsec cert import filename <name of CRL file>
Object #1: CRL
  Issuer: cn=Certificate Manager, o=Thomson, c=BE
  thisUpdate: Sat Jul 28 15:48:51 2002  nextUpdate: Sat Jul 28 16:08:51

Are you sure you want to import this certificate/CRL ? (y/n) :y
```

Note Make sure the time configured at the device is located within the validity time interval of the CRL.

3.2.4 Importation successful?

CRL listing:.

```
[ipsec cert]=>list
Item  Distinguished Name  Type               Issuer    Serial Number

1    cn=Certificate Manager, o=Thomson, c=BE  CRL     Cert #3
2    cn=EndEntityCert, o=Thomson              CERT    Cert #3    113
3    cn=Certificate Manager, o=Thomson, c=BE  CERT    Self-signed 1
```

CRL details:

```
[ipsec cert]=>list item 1
Issuer:  cn=Certificate Manager, o=Thomson, c=BE
   Last Update:  Wed Feb  6 16:50:08 2002
   Next Update:  Wed Feb  6 17:10:08 2002
  Revoked Cert
Serial Numbers:  188, 187, 186, 185, 184, 183, 182, 181, 176, 175, 174,
173, 170, 162, 161, 160, 159, 158, 146, 143, 129, 128, 125, 124, 123,
121, 119, 118, 115, 114, 113, 112, 111, 110, 109, 108, 107, 106, 105,
104, 103, 293, 102, 100, 288, 95, 93, 83, 70, 258, 252, 251, 250, 249,
248, 247, 246, 245, 244, 243, 242, 241, 240, 239, 238, 237, 234, 233,
232, 230, 229, 37, 34, 33, 220, 219, 218, 217, 216, 215, 214, 213, 212,
21, 203, 202, 201, 200, 199, 8, 198, 197, 196, 195, 193, 192, 191
```

3.3 Online CRL retrieval

A lot of configuration options are available in order to retrieve a CRL online. The key options are explained below and for a more detailed explanation, please refer to the appendix.

- Specify the location where to obtain the CRL from:

```
:ipsec cert crlconfig
     dist_point1="ldap://10.11.12.141:389/ cn=Certificate Manager, o=Thomson"
```

- Enable the checking of the CRL:

```
:ipsec cert crlconfig checking_enabled enabled
```

- Enable the dynamic (CEP) retrieval of the CRL

```
:ipsec cert crlconfig fetch_dynamically enabled
```

We get:

```
[ipsec cert]=>crlconfig
CRL Checking   : enabled
Use expired    : disabled
URL            : ldap://10.11.12.141:389/ cn=Certificate Manager, o=Thomson
FetchDynamic   : enabled
Use cert ext   : disabled
Timechecks     : enabled
Net timeout    : 10 seconds
CRL HTTP proxy : disabled
```

If the configuration is done correctly, the CRL will be dynamically retrieved from the CA and imported onto the SpeedTouch^{TM}610.

3.4 CRL logic

- Enable CRL checking
  By default, CRL checking is disabled. The command below activates CRL checking:

```
:ipsec cert crlconfig checking_enabled enabled
```

- Check certificate revocation
  By listing the details of both the end-entity certificate and CRL, it is possible to verify whether the end-entity certificate is revoked:

```
[ipsec cert]=>list item 1
Issuer:  cn=Certificate Manager, o=Thomson, c=BE
   Last Update:  Wed Feb  6 16:50:08 2002
   Next Update:  Wed Feb  6 17:10:08 2002
  Revoked Cert
Serial Numbers:  188, 187, 186, 185, 184, 183, 182, 181, 176, 175, 174,
173, 170, 162, 161, 160, 159, 158, 146, 143, 129, 128, 125, 124, 123,
121, 119, 118, 115, 114, 113, 112, 111, 110, 109, 108, 107, 106, 105,
104, 103, 293, 102, 100, 288, 95, 93, 83, 70, 258, 252, 251, 250, 249,
248, 247, 246, 245, 244, 243, 242, 241, 240, 239, 238, 237, 234, 233,
232, 230, 229, 37, 34, 33, 220, 219, 218, 217, 216, 215, 214, 213, 212,
21, 203, 202, 201, 200, 199, 8, 198, 197, 196, 195, 193, 192, 191


[ipsec cert]=>list item 2
        Status:  Revoked
       Version:  V3
        Issuer:  cn=Certificate Manager, o=Thomson, c=BE
     Serial No:  113
 Validity Date:  not before: Fri Jul 27 12:45:49 2001
                 not after:  Sat Jul 27 12:45:49

       Subject:  cn=EndEntityCert, o=Thomson
    Extensions:  _keyUsage: digitalSignature, keyEncipherment,
                 _authorityKeyIdentifier
                 _CRLDistributionPoints
          1. cn=Certificate Manager, o=Thomson

[ipsec cert]=>
```

Note As the serial number of the certificate is listed in the CRL, the state of the certificate is reflected as "revoked".

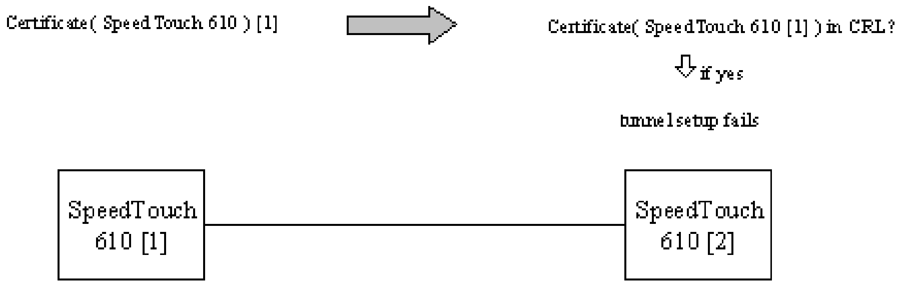

Certificate(SpeedTouch 610) [1]
Certificate(SpeedTouch 610 [1]) in CRL?
if yes
tunnel setup fails
SpeedTouch 610 [1]
SpeedTouch 610 [2]

Figure 6: CRL checking on remote IPSec peer

During tunnel setup, the remote IPSec peer will verify the certificate validity of the other peer. If the certificate is revoked, authentication will fail and no secure tunnel will be setup between the two SpeedTouch^{TM}610 devices. Take into account that only the certificate of the remote IPSec peer is verified, not the local certificate.

When CRL checking is enabled and no CRL is imported into the secure storage of the SpeedTouch^{TM}610, all certificates in the secure storage are considered invalid.

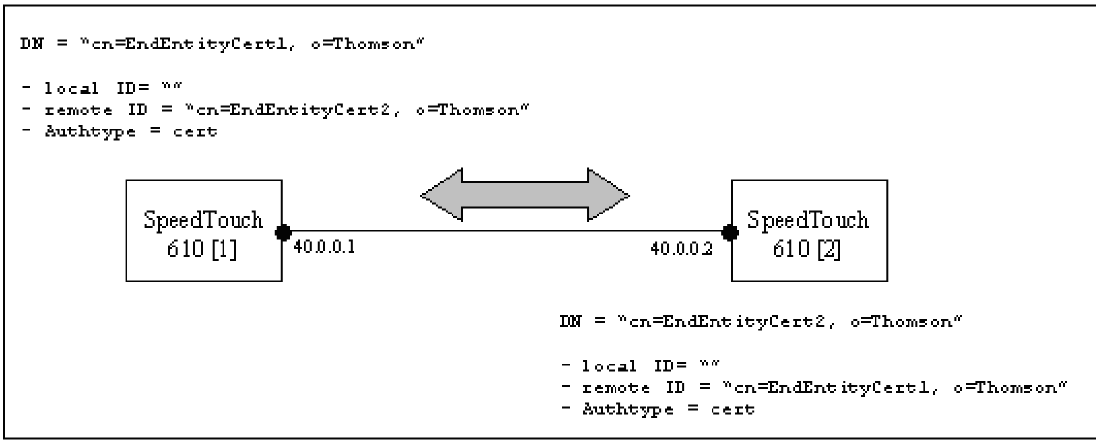
4 NEGOTIATION DURING CERTIFICATES

Below a basic configuration example is listed. For more details on the IPSec peer configuration, please refer to the application note “SpeedTouch^{TM}610 IPSec configuration: demystified”.

To configure the SpeedTouch^{TM}610 to use certificates for authentication, enter following parameters for the IPSec peer:

- Authentication type: enter “cert”
- Remote ID: fill in the distinguished name of the remote IPSec peer’s certificate (optional).

EXAMPLE 1



```
DN = "cn=EndEntityCert1, o=Thomson"

- local ID= ""
- remote ID = "cn=EndEntityCert2, o=Thomson"
- Authtype = cert
```

```
DN = "cn=EndEntityCert2, o=Thomson"

- local ID= ""
- remote ID = "cn=EndEntityCert1, o=Thomson"
- Authtype = cert
```

Figure 7: Configure SpeedTouch^{TM}610 for certificates

:

	SpeedTouch^{TM}610 [1]	SpeedTouch^{TM}610 [2]
Peer	cert_peer_SpeedTouch_1	cert_peer_SpeedTouch_2
Address	40.0.0.2	40.0.0.1
Local ID	““””	““””
Remote ID	cn=EndEntityCert2, o=Thomson	cn=EndEntityCert1, o=Thomson
Authtype	cert	cert
Secret		
Descriptor	Phase1Descr	Phase1Descr

Note As the DN of the local certificate is available in the local secure storage, it makes no sense entering the “Local ID” parameter.

EXAMPLE 2

It is not mandatory to fill in the remote ID (see below). However, when filling in the remote ID, it must have a perfect match.

	SpeedTouch^{TM}610 [1]	SpeedTouch^{TM}610 [2]
Peer	cert_peer_SpeedTouch_1	cert_peer_SpeedTouch_2
Address	40.0.0.2	40.0.0.1
Local ID	““	““
Remote ID	““	““
Authtype	cert	cert
Secret		
Descriptor	Phase1Descr	Phase1Descr

5 MODIFY SECURE STORAGE

To reload/load the certificates and CRLs found in the secure storage into memory:

```
:ipsec cert refresh
```

To remove one item from the secure storage:

```
:ipsec cert remove
```

Note When removing the CA certificate, all certificates in the secure storage being signed by this CA get listed as invalid.

To remove all items from the secure storage (i.e. clear the secure storage):

```
:ipsec cert clearall
```

Note To make sure all certificate information is wiped out, the below procedure needs to be followed:

```
=>ipsec cert clearall
=>software deletepassive
=>software duplicate
=>software switch
=>
```

An individual certificate can be exported and re-imported:

```
:ipsec cert export
:ipsec cert import
```

Use this command for offline certificate import. The signed certificate – in PKCS#7 format – is first placed onto the devices filesystem using FTP and then the importation can be done. These commands allow for transferring certificates between different SpeedTouch^{TM}610 devices.

Export example:

```
=>ipsec cert export item=1 filename=test.cert
```

Use an FTP session to retrieve the exported certificate file from the SpeedTouch^{TM}610 (with IP address 10.0.0.138):

```
C:\>ftp 10.0.0.138
Connected to 10.0.0.138.
220 Inactivity timer = 120 seconds. Use 'site idle <secs>' to change.
User (10.0.0.138:(none)): admin
331 SpeedTouch (00-90-D0-01-88-2D) User 'admin' OK.  Password required.
Password:
230 OK
ftp> bin
200  TYPE is now 8-bit binary
ftp> cd dl
250 Changed to /dl
ftp> get test.cert
200 Connected to 10.0.0.1 port 2141
150 Opening data connection for test.cert (26)

226 File transfer complete
ftp: 26 bytes received in 0.01Seconds 2.60Kbytes/sec.
ftp>bye
221 Goodbye.  You uploaded 0 and downloaded 1 kbytes.

C:\>
```

Use an FTP session to put the retrieved certificate file to another SpeedTouch^{TM}610 (with IP address 10.0.0.139):

```
C:\>ftp 10.0.0.139
Connected to 10.0.0.139.
220 Inactivity timer = 120 seconds. Use 'site idle <secs>' to change.
User (10.0.0.139:(none)): admin
331 SpeedTouch (00-90-D0-01-88-2A) User 'admin' OK.  Password required.
Password:
230 OK
ftp> bin
200  TYPE is now 8-bit binary
ftp> cd dl
250 Changed to /dl
ftp> put test.cert
200 Connected to 10.0.0.1 port 1251
150 Opening data connection for test.cert
226 File written successfully
ftp: 26 bytes sent in 0.00Seconds 26000.00Kbytes/sec.
ftp> bye
221 Goodbye.  You uploaded 1 and downloaded 0 kbytes.

C:\>
```

Import example:

```
=>ipsec cert import filename test.cert

Object #1: Certificate
  Subject: cn=nat1, o=Thomson
  Issuer: cn=Certificate Manager, o=Thomson, c=BE
  notBefore: Tue Jan  8 13:10:17 2002  notAfter: Wed Jan  8 13:10:17 2003

Are you sure you want to import this certificate/CRL ? (y/n) :y
=>
```

Note When installing the EE certificate “test.cert”, make sure the issuer certificate is already imported.

6 TROUBLESHOOTING CERTIFICATES

6.1 Checking Validity of Certificates

Certificates have only a limited lifetime and once expired, the certificate becomes invalid. Check validity with the below command:

```
:ipsec cert list item=<x>
```

Make sure the time on the device is set correctly. After rebooting the device, time settings have to entered again using the below command:

```
:system settime date 2/2/2003 time 12:23
```

We can configure an SNTP (Simple Network Time Protocol) server in order to retrieve the time from a remote time server. To configure an SNTP server:

```
=>sntp config enable yes
=>sntp add addr=192.168.1.2 version=3
=>sntp list
IP Address      Version   Status
192.168.1.2     3         synchronized
=>
```

There is a configuration option that allows the time checking of the certificates to be disabled: even if the certificate is expired, a secure tunnel can be setup. This option is introduced for users having difficulties with retrieving the correct time. The user is not encouraged to use this option as it compromises security.

```
:ipsec cert crlconfig time_checking=off
```

Check whether the certificate is listed in the CRL.

6.2 Certificates States

Using the ":ipsec cert list item <x>" command, the state of the certificate can be retrieved. Following states are supported:

- valid
- not valid yet
- revoked
- expired
- update required (set when <20% lifetime is left on the certificates validity date)

A syslog error message will report the expiry of a certificate, e.g. when your certificate is going to expire you get:

```
<3> SysUpTime: 01:14:37 VPN : There is less than 20 percent of the certificates lifetime
remaining.  Please update your certificate.
```

7 CLI COMMANDS SUMMARY

7.1 CEP configuration options

- [url = <quoted string>]: the CEP enrollment URL.
  The URL of the CEP script on the CA server. It's usually in the form http://<host>[:<port>]/cgi-bin/pkiclient.exe. This entry must have a value, e.g.: "EnrollmentURL=http://192.168.1.2:12000/cgi-bin/pkiclient.exe".
  - [ip address] is a numeric address, do not enter a DNS name
  - [port] is the port number (optional)
  - [path] is the path to the CA's CEP script
- [ca_id = <quoted string>]: CA Identity string.
  The CAIdentity string gives the id of the CA server as some PKIs use a string to identify a CA.
- [md5 = <quoted string>]: CA cert. MD5 fingerprint in hexdump format (0A:CB:9F ...).
  The CAMD5FingerPrint is the finger print (MD5 hash) of the root CA's certificate. MD5 hash in hex, must be 16 bytes long and bytes are separated with ":". WARNING: the root CA will not be authenticated if this entry is not set. E.g.: CAMD5FingerPrint=92:EC:0F:CD:51:DF:29:87:DB:3F:0B:68:5E:A4:A2:99.
- [proxy_url = <quoted string>]: proxy server URL.
  Represents the IP address of the http-proxy-server and is required if the CA is located behind a firewall. The value of this entry is in the form <host>[:<port>]. If port is not set, the standard HTTP port number will be used. E.g.: ProxyServer=192.168.1.2:8080.
- [subjectdn = <Distinguished Name (RFC1779): e.g. cn=xxx, c=be>]: distinguished name (RFC1779) for the cert.
  Reflects the subject name for the requested certificate. The value of this entry must be a valid distinguished name in string representation (RFC 1779). The SubjectDN can include common name (cn=), organization unit (ou=), organization name (o=), locality (l=), province or state (st=), and country (c=). Use commas to separate the items, and enclose all items in quotation marks.
- [keylen = <number>]: keylength in bits.
  The key length is an RSA compliant key length (i.e.: 512, 1024 or 2048). All you can do to test this is to create different certificates with each size. Provided you CA supports those key sizes everything should just work (the key is, essentially, your certificate). So keylength is the length of the being certified key and only RSA keys are supported. Default is 1024.
- [password = <password>]: the challenge word (if needed).
  If your CA requires a password for revocation or automatic enrollment, this is where you set it up. It is really an infrastructure specific setting. Some PKI's may exploit XAuth for passwords while others don't use passwords at all. It is typically a challengepassword used by the CA to authenticate the enrollment request or the revocation request. Check that this password is not visible in the clear (not in the traces nor in the content of the packets).
- [email = <quoted string>] | [dnsname = <quoted string>] | [ipaddress = <ip-address>] | [altsubjectdn = <Distinguished Name (RFC1779): e.g. cn=xxx, c=be>]
  "EmailAddress", "DnsName", "IPAddress", and "AltSubjectDN" are entries used to form a SubjectAltName X509v3 extension in the requested certificate. See below for more details.

Note When you set the several options in the "X509v3 extension" of the CEP, it is possible you don't see them anywhere appearing on the CA or the cert list. This is because the inclusion of these options is dependant on the way the CA is set up. It may ignore/include/add options as it sees fit. The best way to describe these options is to see them as "suggested" extensions, not hard requirements.

  - [email = <quoted string>]: email address (RFC822) for X509v3 extension.
    You don't need it, it's optional. It is used to create the certificate if present. Emails are not sent anywhere and a server is not required to be installed. So the signed CEP request is not send automatically to this email address.
  - [dnsname = <quoted string>]: domain name for X509v3 extension.
    Again, this is optional. Some customers may want to give their SpeedTouch 610 a real DNS name.
  - [ipaddress = <ip-address>]: IP address for X509v3 extension.
    Same as above. Some customers may want to associate the IP address.
  - [altsubjectdn = <Distinguished Name (RFC1779): e.g. cn=xxx, c=be>]: distinguished name (RFC 1779) for X509v3 extension.
    The SubjectAltName is a certificate extension that is used to help further identify the user of the certificate. See RFC 2459 Section 4.2.1.7 for more details. It is _not_ something like this DN name being chosen when the other DN (subjectdn) name is rejected.
- [chknonce = <{yes|no}>]: enable/disable nonce checking. ("no" for Entrust CA).
  Required for interoperability reasons. This flag is disabled because Entrust does not properly conform to the CEP spec. If not disabled, the nonce check (part of CEP) will always fail because Entrust does not use it correctly... Default should be "Yes".
- [chktid = <{yes|no}>]: enable/disable transaction Id checking. ("no" for Baltimore CA).
  Same as nonce checking. This is required for Baltimore interoperability. They do not do transaction id checking. If this is not disabled for Baltimore, we will never work. Case insensitive. Default should be "Yes".
- [keyusage = <quoted string>]: the key usage extension. Format : [yes/no]
  This option allows you to specify the keyusage extensions required. See RFC 2459 for an detailed explanation of the keyusage extension. Typically you won't care to use this as this entry is not required for most CAs.
  The key usage extension entry should be in the form <critical>, <usage1>, <usage2>..., where <critical> can be YES or NO. The error message "SCEP parameter syntax error:" should be logged if <critical> is missing.
  An example would look something like: "keyUsage = no, digitalSignature, nonRepudiation".

[digitalSignature,nonRepudiation,keyEncipherment,dataEncipherment,keyAgreement,keyCertSign,cRLSign,encipherOnly,decipherOnly], [ ] ...

Only applicable when the keyusage flag is set. The default parameters are set by the CA. See table below for the available options:

Key usage	Description
digitalSignature	Digital signatures authenticate the source of data. Your private key is applied to all outgoing communication, creating a digital signature. A receiving party verifies that you are the sender by decrypting the information using your public key, which they obtain from the CA.
nonRepudiation	If the signing party performs some action (for example, issuing or receiving specific data), non-repudiation prevents denial of that action. P.S.: non-repudiation means the inability, of the sender of a message, to deny having sent the message. A regular hand-written signature provides one form of non-repudiation.A digital signature provides another.
keyEncipherment	Key encipherment ensures the confidentiality of your keys.
dataEncipherment	Data encipherment ensures data confidentiality.
keyAgreement	Key agreement refers to the generation of new shared keys between two parties.
keyCertSign	Certificate signing refers to the verification of the signature on a CA's certificate.

Key usage	Description
cRLSign	Revocation information signing refers to the verification of the signature on revocation information.
encipherOnly	When establishing key agreement, the public key is only used to encipher data.
decipherOnly	When establishing key agreement, the public key is only used to decipher data.

7.2 CRL configuration options

- [checking_enabled = <{yes|no}>]
  Indicates whether certificate revocation checking is performed. Options are Yes and No, where Yes enables and No disables revocation checking.
- [use_expired_crls = <{yes|no}>]
  Used to determine whether to use a CRL if it is expired. “Expired” refers to a CRL that has passed its “nextUpdate” field. For further details refer to RFC 2459 (Internet X.509 Public Key Infrastructure Certificate and CRL Profile). Options are YES and NO, where YES allows the SpeedTouch 610 to continue to use expired CRLs when verifying a certificate and NO does not. If set to NO and the gates CRL expires and the CRL is not updated, all further SA requests will fail, until the CRL is updated.
- [dist_point1 = <quoted string>]
  The URL/URI to fetch CRL's from. It indicates the location a CRL should be retrieved from. If “CRL_FETCH_DYNAMICALLY” is enabled, this parameter must be present. The values must be in the form of a URI and the supported protocols include LDAP and HTTP. The server name portion of the distribution point should be in the form of an IP address. Refer to RFCs 1738, 1779 and 1957 for further details on URIs, DNs and LDAP URIs respectively. Examples are:

  ```
  CRL_DIST_POINT1 = ldap://1.1.1.6:17002/cn=MasterCRL, ou=crlIssuingPoints, ou=ca,
  o=netscapeCertificateServer

  CRL_DIST_POINT1 = http://172.23.99.11:17002/repository/crl.p7b
  ```

- [fetch_dynamically = <{yes|no}>]
  Determines whether to attempt to fetch a CRL automatically. Options are Yes and No, where Yes means that the gate will attempt to fetch a CRL from the distribution point identified by “CRL_DIST_POINT1”. No disables automated CRL retrieval. If the "CRL_DIST_POINT1" parameter is not set up properly, CRL fetching will not be enabled.
- [check_cert_extension = <{yes|no}>]
  Description: determines whether to check the Certificate extension “crlDistributionPoints” for a valid CRL distribution point. Options are Yes and No, where Yes enables the check and No disables this check. If enabled the certificates do not need to have this extension in order to retrieve a certificate. The SpeedTouch 610 will default back to the value of "CRL_DIST_POINT1".
  As the cert_ext_usage parameter uses DNS names and the SpeedTouch^{TM}610 only accepts IP addresses, this option will only work if the full IP address is present in the extension.

To check what the CRL retrieval point is:

```
[ipsec cert]=>list item 2
        Status:  Valid
       Version:  V3
        Issuer:  cn=Certificate Manager, o=Thomson, c=BE
     Serial No:  113
 Validity Date:  not before: Fri Jul 27 12:45:49 2001
                 not after:  Sat Jul 27 12:45:49

       Subject:  cn=EndEntityCert, o=Thomson
    Extensions:  _keyUsage: digitalSignature, keyEncipherment,
                 _authorityKeyIdentifier
                 _CRLDistributionPoints
          1. ldap://138.203.14.183:389/cn=Certificate Manager, o=Thomson
```

Below the decision logic to choose the correct LDAP distribution point:
The cert extension has priority. If it is an incomplete distribution point (meaning the server and port information is missing), then the default (the one present in the manually entered distribution point-param.) is used to build the new distribution point. If the cert. extension is not successful, the distribution point is used.

Note Even if the LDAP location is completely specified in the certificate extension, still a user needs to enter manually a string for the LDAP distribution point (see parameter "dist_point1").

Example:

```
[ipsec cert]=>list item 2
       Version:  V3
        Issuer:  cn=Certificate Manager, o=Thomson, c=BE
     Serial No:  331
 Validity Date:  not before: Thu Mar  7 13:57:03 2002
                 not after:  Fri Mar  7 13:57:03 2003

       Subject:  cn=NewModemTest, o=Thomson
    Extensions:  _keyUsage: digitalSignature, keyEncipherment,
                 _authorityKeyIdentifier
                 _CRLDistributionPoints
                  1. CN=Certificate Manager,O=Thomson
```

Note Notice the incomplete distribution point (DP) in the cert. extension.

```
[ipsec cert]=>crlconfig
CRL Checking   : enabled
Use expired    : enabled
URL            : ldap://138.203.14.183:377/CN=another location, O=Thomson
FetchDynamic   : enabled
Use cert ext   : enabled
Timechecks     : enabled
Net timeout    : 10 seconds
CRL HTTP proxy : disabled
```

Note Look at the different location between the certificate extensions crlDistributionPoint and the default DP.

Because the crlDistributionPoint is an incomplete location (no server/port) we use the default DPs to create the following DP:

```
 ldap://138.203.14.183:377/CN=Certificate Manager,O=Thomson
```

If this fails, we will fall back to the default DP of:

```
ldap://138.203.14.183:377/CN=another location, O=Thomson
```

- Timechecks
  Disables the time checking on your certificate, thus allowing for the use of expired certificates.
  Reason of this feature: it can be an issue for the user to retrieve the correct time (currently the 610 has no real-time clock, or the SNTP server maybe be unreachable,...). Therefore a configuration option is available allowing to disable the validity checking of a certificate. Take into account that this is a security issue (expired certificates can be used), but this a policy decision is to be taken by the user and it can be a valuable work-around.

Note Although an expired CRL can be used when disabling this option, it is not possible to import an expired CRL. So first import a non-expired CRL, then disable the timecheck-option and from now on the CRL will be used disregarding its expiry.

- Net timeout
  Applies to both online CEP and CRL retrieval. Specifies the number of seconds to wait for the answer of the CA before cancelling the request.
- CRL HTTP proxy
  Enter <IP address HTTP server>:<port on which the HTTP server is listening>. The IP address and port number point to the location from where the CRL can be retrieved.

www.speedtouch.com

Acknowledgements

All Colleagues for sharing their knowledge.

Coordinates

THOMSON

Prins Boudewijnlaan 47
B-2650 Edegem
Belgium

Email: documentation.speedtouch@thomson.net

Copyright