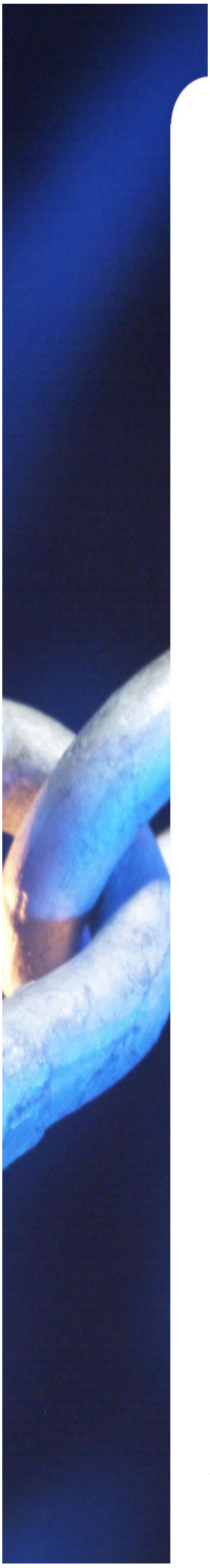

ipSEC EXPRESS

SSH X.509 Certificate Tools

White Paper

Version 2.0, March 1999

ssh COMMUNICATIONS SECURITY

SSH X.509 Certificate Tools

White Paper

Version 2.0, November 1999

http://www.ssh.fi/

SSH Communications Security Ltd.
Tekniikantie 12
FIN-02150 Espoo
Finland

http://www.ssh.fi
E-mail: ipsec-sales@ssh.fi
Tel: +358-9-4354 3208
Fax: +358-9-4354 3222

Table of Contents

Summary

The SSH X.509 Certificate Tools is a complete library for applications that require strong authentication of parties through digital certification. With the SSH X.509 Certificate Tools, an application can connect to a Public Key Infrastructure (PKI) and use the certification services of a certificate authority for maximum security.

The SSH X.509 Certificate Tools can be used e.g. to retrieve certificates from a public Certification Authority (CA) through directory access, for processing Certificate Revocation Lists (CRLs) and for generating certificate requests. SSH X.509 Certificate Tools library has been thoroughly tested for interoperability against the products of all major CA vendors on the market. It is designed to integrate easily to the products of OEMs to add certificate management functionality into their own products.

Award winning IPSEC solutions by SSH Communications Security are being used in multiple locations and countries around the world. The SSH X.509 Certificate Tools library forms one important module of SSH IPSEC Express and SSH ISAKMP/Oakley (SSH IKE) toolkits. It can be used as part of SSH IPSEC Express and SSH IKE products or as a separate certificate management product.

A Full Set of Tools for Certificate Processing

The SSH X.509 Certificate Tools has been designed to be an easy-to-use out of the box certificate processing library and tool. It includes many ready-made programs for creating and processing certificates with on-line commands and the simple API that can be used to further integrate the library into any application requiring it.

Everything required exits in the library, including the processing routines, directory access protocols and functions, cryptography, and a fast mathematics library.

Applications

Applications for the SSH X.509 Certificate Tools include

- Third party IPSEC implementations in VPNs, routers, firewalls, etc.
- Cryptographic protocol implementations
- Electronic commerce applications
- Other security aware applications that need to assume the security of the hosts it is running in or communicating with.

X.509 Certificate Processing

The SSH X.509 Certificate Tools support X.509v3 certificates. These tools make it possible to:

- Verify the X.509v3 certificates and add public keys to the cache or database.
- Generate X.509v3 certificates.
- Generate PKCS #10 certificate requests.
- Verify certificate chains up to a trusted Certificate Authority (CA).
- Check X.509 CLRs.
- Use LDAP for fetching certificates and CRLs.

All Required Cryptography Included

The SSH X.509 Certificate Tools comes with the SSH Cryptographic Library, which is a full set of strong cryptographic algorithms. There is not need for a separate cryptographic library. Designed and implemented by cryptography specialists at SSH Communications Security, The SSH Crypto Library is a very efficient and secure implementation of the most secure and commonly used algorithms on the market

Introduction to Certificate Technology

Several different methods exist for authenticating two or more parties that need to communicate over insecure networks such as the Internet. Such methods include, for example, passwords, challenges, and shared secrets. However, the flexibility and security of the authentication can be greatly improved by introducing third parties. These third parties are trusted by everyone to certify the true identity of the parties who have perhaps never met, prior to the intended communication. These third parties are called certificate authorities (CAs) and the system for the authentication service is called a Public Key Infrastructure.

Now, with the emergence of electronic commerce and legislation to govern digital identification and signatures as valid instruments for identification, the PKIs are fast spreading to authenticate all actors in these networks, including both people and machines. Those hosts on the networks that take care of mission critical encryption of the confidential data need to be authenticated as well as the persons. This can be also accomplished through digital certification. To take the advantage of this service, the applications need an unbreakable method for connecting to those systems. The SSH X.509 Certificate Tools is just what is needed.

Certificates and Certificate Requests

Certificates are digital documents of identification. As real life identification documents, they have a set of attributes that define the owner of the certificate. A third party official has also granted them. The attributes of the certificate have been encoded in the certificate according to a standard. The X.509 is one such standard.

The X.509 certificates and CRLs are defined using the abstract syntax notation one (ASN.1) and distinguished encoding rules (DER) to convert them to binary. The SSH X.509 Certificate Tools transparently handles these issues. Application doesn't need to work its way through the smallest details but can directly utilize certificates and CRLs.

The library includes a C API, which gives easy access to the elements of certificates, and CRLs. For more high-level operations we give also a script forms for generating certificates, CRLs and PKCS 10 certificate requests.

Basically certificates consist of names, for the subject and the issuer, a public key for the subject, and the signature. The CRLs contain basically name of the CA and list of certificate serial numbers to identify certificates revoked.

The script form allows straightforward construction of certificates, in a way closely following the ASN.1 descriptions, but still allowing easy use.

Figure 1 illustrates the structure of a simple certificate form.

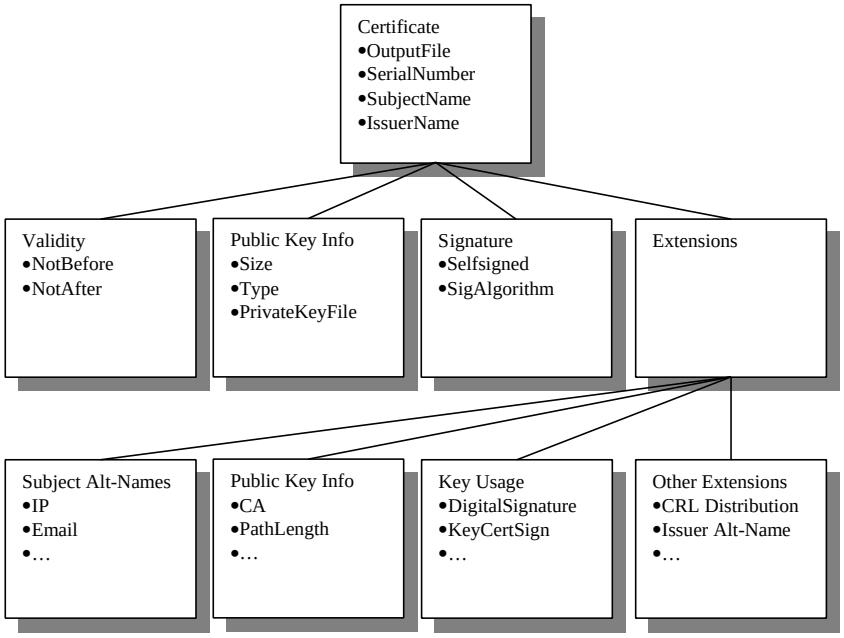

Figure 1. Simplified hierarchical structure of Certificate

In text below is explained some of the most common terms of the variables used in the certificate form:

- OutputFile. The certificate file to which the generated certificate in raw binary format..
- SerialNumber. The serial number is a unique identifier for the certificate under the signing Certificate Authority. Serial number and issuer name can be uniquely used to revoke a certificate.
- SubjectName. In practice subject name fields describes the X.500 distinguished name. The structure of a distringuished name is a hierarchial one, often following some X.500 directory structure. The name consist of many parts, some of which are: country (e.g. "C=FI"), organization name (e.g. "O=SSH Communication Security"), organization unit (e.g. "OU=Marketing"), common name of the subject (e.g. "CN=Johnny Smith").
- IssuerName. As above, thus equals the distinguished name of issuer.
- Subject Alternative Names. This extension is used for alternative naming methods for subject. This may consist of the subject, the Domain Name Server (DNS) name of the subject (e.g. "ssh.fi") and email address of the subject (e.g. "info@ssh.fi").
- PrivateKeyFile. The name of the file where the subject's private key will be stored in raw binary format.
- Type. The type of the key to be produced. The possibilities are rsaEncryption (RSA keys. Suggested sizes are above or equal to 1024 bits) and dsaEncryption (DSA keys. Suggested size is 1024 bits, which is the largest allowed by NIST. However, one can set this to larger if needed)
- Size. Size of the key in bits.

- SignatureAlgorithm. The signature algorithm to be used. The name should be a valid X.509 signature algorithm name. E.g. md5WithRSAEncryption or dsaWithSHA-1.
- NotBefore. The date since when the certificate is valid.
- NotAfter. The date after when the certificate is invalid.
- CA. Defines if the certificate is a CA certificate or not.
- PathLength. Defines the maximum path length allowed in certificate lists certified under this certificate.

Figure 2 illustrates what a typical certificate form looks like

```
Certificate ::= {
  OutputFile   ::= "test-ca-1.bin"

  SerialNumber ::= 0
  SubjectName  ::= <C=FI,O=SSH Communications Security\, Ltd., CN=Test CA>
  IssuerName   ::= <C=FI,O=SSH Communications Security\, Ltd., CN=Test CA>
  Validity     ::= {
    NotBefore  ::= "1998 Jul 30th, 19:30:00"
    NotAfter   ::= "2000 Jan 1st,  12:00:00"
  }
  PublicKeyInfo ::= {
    Size ::= 1024
    Type ::= rsaEncryption
    PrivateKeyFile ::= "test-ca-1.prv"
  }
  Signature ::= {
    SelfSigned
    SignatureAlgorithm ::= md5WithRSAEncryption
  }
  Extensions ::= {
    SubjectAltNames ::= {
      IP    ::= 1.2.3.4
      EMAIL ::= test-ca@ssh.fi
    }
    BasicConstraints ::= {
      CA
      PathLength ::= 0
    }
    KeyUsage ::= {
      DigitalSignature
      KeyCertSign
    }
    CRLDistributionPoints ::= [
      {
        FullName ::= {
          DN ::= <C=FI,O=SSH Communications Security\, Ltd.,CN=CRL1>
        }
        CRLIssuer ::= {
          DN ::= <C=FI,O=SSH Communications Security\, Ltd.,CN=Test CA>
        }
        KeyCompromise
      }
      {
        IssuerRelativeDN ::= <C=FI,O=SSH Communications Security\, Ltd.,CN=Test CA,CN=CRL1>
        CRLIssuer ::= {
          DN ::= <C=FI,O=SSH Communications Security\, Ltd.,CN=Test CA>
        }
        CaCompromise
        AffiliationChanged
        Superseded
        CessationOfOperation
        CertificateHold
      }
    ]
  }
}
```

Figure 2 An example certificate form

Certificate requests are the method used for requesting certificates from a CA. There exists a standard for this and it is called PKCS#10. The SSH X.509 Certificate Tools support PKCS#10. Again, using the simple form structure one can easily generate such a request.

Figure 3 illustrates the structure of a certificate request form**.**

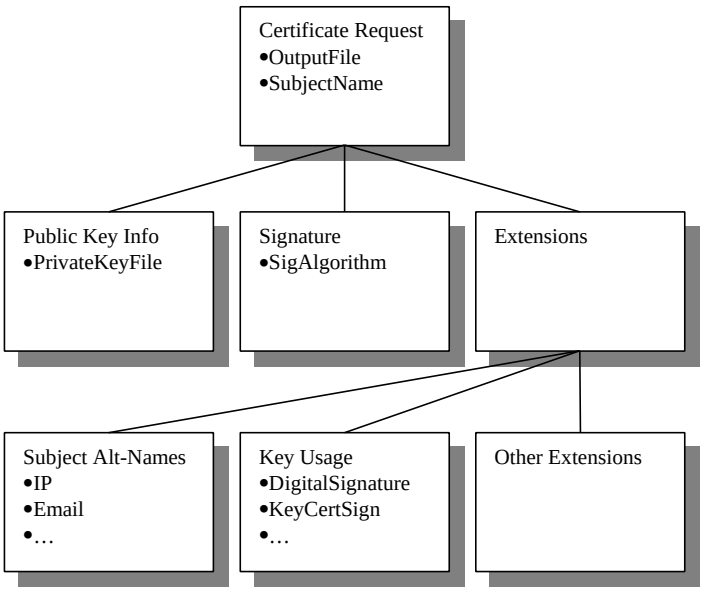

Figure 3. Structure of a certificate request form

The variables are very much the same as in a certificate. However, a couple of them deserve more attention:

- OutputFile. The request output file the PKCS 10 certificate request is to be stored in raw binary format.
- SignatureAlgorithm. The signature algorithm used for self-signing.
- PrivateKeyFile. The file where the private key is stored.

Figure 2 illustrates what a typical certificate form looks like

```
CertificateRequest ::= {
  OutputFile   ::= "test-user-1.req"
  SubjectName  ::= <C=FI,O=SSH Communications Security\, Ltd., CN=Test User 1>
  PublicKeyInfo ::= {
    InputPrivateKeyFile ::= "test-user-1.prv"
  }
  Signature ::= {
    SignatureAlgorithm ::= md5WithRSAEncryption
  }
  Extensions ::= {
    SubjectAltNames ::= {
      IP ::= 192.168.2.4
    }
    KeyUsage ::= {
      DigitalSignature
      KeyEncipherment
    }
  }
}
```

Figure 4 An example certificate request form

Using forms for certificate and certification generation is widely deployed and perhaps the easiest way of doing things, but not the only way. SSH X.509 Certificate Tools also includes a complete C source code libraries and detailed technical documentation for using certificates as embedded C source code almost just as easily.

Directory Services and Certificate Revocation

X.500 directories are a means of storing certificates in a PKI system based on X.509 certificates. In case an application needs the certificate of a party it is willing to communicate with and has not obtained that certificate from the party, it can get the certificate from a directory of a CA it trusts.

The CA has a database of certificates and certificate revocation lists (CRLs) that can be accessed with directory access. A popular method for the retrieval of CRLs and certificates is the Lightweight Directory Access Protocol (LDAP), which is included in the SSH X.509 Certificate Tools. The LDAP client connects to the directory and retrieves the required certificate based on a key word, which is a field in the certificate.

Once retrieved, the client can check the validity of the certificate by using the public key of the CA. The validity period is written on the certificate. However, in case the certificate has to be revoked before the original expiration, this revocation can be communicated to the parties via a certificate revocation list, which lists the revoked certificates.

The SSH solution also supports certificate manager, which gives transparent interface to the control of certificates and CRLs. Simply by using this any application can have LDAP searches for certificates and CRLs happen automatically. The application does not have to concern itself with complex details, such as validity periods or timely revoking of certificates.

SSH Certificate Manager (CM) interfaces gives application full control of X.509 certificates and CRLs to such a degree that the application can concentrate on high-level policy issues instead of intricate details of certificate management.

SSH X.509 Certificate Tools as Part of SSH IPSEC Express

SSH IPSEC Express is a complete, easy-to-use toolkit for adding IPSEC functionality into a product of a third party. The SSH X.509 Certificate Tools is a module included in SSH IPSEC Express package to greatly enhance the IPSEC features available. One part of SSH IPSEC Express is SSH ISAKMP/Oakley (SSH IKE), which is a full implementation of the ISAKMP/Oakley protocol. The SSH X.509 Certificate Tools integrate seamlessly into the IKE key management and can be used for easy integration of the IPSEC solution to different PKIs. For more information about the SSH IPSEC express, please read the SSH IPSEC Express White Paper that can be found from the SSH Web site at http://www.ssh.fi/tech/whitepapers.html.

Interoperability

The interoperability of SSH X.509 Certificate Processing Tools has been tested with the products of the leading CA vendors. The SSH X.509 Certificate Processing Tools is fully interoperable with the products of at least the following companies

- Baltimore Technologies
- Entrust
- VeriSign
- XCert
- AU Systems (ID2 Technologies)
- Hi/fn
- Microsoft

Conclusions

IPSEC is a technology that will solve many of the security problems of the current Internet. It has a wide support from all the major players in the industry. Certificates greatly enhance the security and services offered by the IPSEC implementations. Additionally, many new applications are being developed based on the services offered by Certificate Authorities. The SSH X.509 Certificate Tools is an ideal library for adding such functionality into any IPSEC implementation or other security aware applications.

Related literature

PKIX draft standards at www.ietf.org/ids.by.wg/pkix.html

S. Kent, R. Atkinson, *"Security Architecture for the Internet Protocol"*, Internet Draft. 1997.

NIST, FIPS PUB 180-1: Secure Hash Standard. April 1995.

Denning, Dorothy E. R. "Cryptography and Data Security", Addison Wesley. 1983.

Halsall, Fred. "*Data Communications, Computer Networks and Open Systems"*, 4th ed, Wokingham, Addison-Wesley. 1995.

Menezes, Alfred J.; van Oorschot, Paul C.; Vanstone, Scott A.." *Handbook of Applied Cryptography"*, New York, CRC Press. 1997.

Schneier, Bruce. "*Applied Cryptography - Protocols, Algorithms and Source Code in C"*, 2nd ed. New York, John Wiley & Sons. 1996

Terms and Abbreviations

This section provides definitions for key terms that are used in this document.

ASN.1

The ASN.1 (Abstract Syntax Notation One) is the standard for describing X.509 structures. The notation is abstract in the sense that it specifies no details how the actual data is to be encoded, for example in computer systems.

BER

The BER (Basic Encoding Rules) describes one possible way of converting structures of ASN.1 into binary data. These objects are not necessary unique in binary form.

CA

Certificate Authority. An entity that attests to the identity of a person or an organization. A CA might be an external company that offers certificate services or it might be an internal organization such as a corporate Management Information System (MIS) department. The chief function of the CA is to verify the identity of entities and issue digital certificates attesting to that identity.

Certificate

Certificate, here X.509 certificate structure, is defined in ASN.1 in the X.509 standard.. Lately, in PKIX draft, standard several useful extensions were defined for X.509v3, which is supported with tools in this document.

CRL

Certificate Revocation List. A list that states, which certificates are not valid anymore.

DER

The DER (Distinguished Encoding Rules) describes another way of converting structures of ASN.1 into binary data. DER is a subset of BER and does guarantee unique representation in binary for every ASN.1 structure

Distinguished Name

The distinguished name is an object giving a distinguished, or unique, name for each object in some namespace. Usually supplied by a CA, and determined by the particular X.500 directory structure.

PEM File Format

PEM (Privacy Enhanced Mail) format is based on base-64 encoding and headers which describe the blob type.