Secure Commnication Made Easy

Public Key Infrastructure (PKI) Tool

PKI_Tool is a standalone GUI application for Windows that implements all functions of a Public Key Infrastructure (PKI). It purpose is to create and manage X.509 digital certificates. It is normally deployed as one component for our *Domain Identity Registry* product

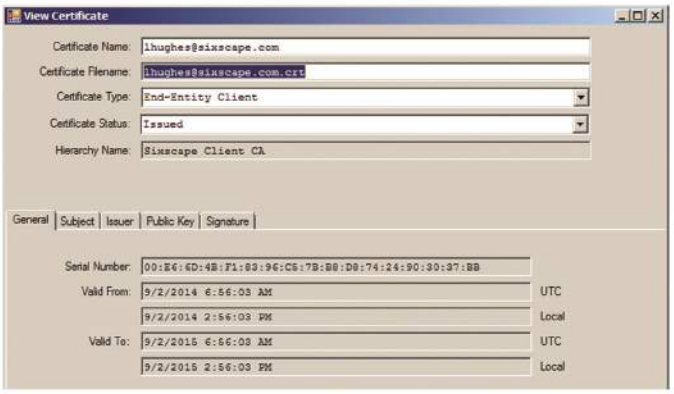
It can create two kinds of digital certificates:
Server Certs and **Client Certs**.

Server Certs bind the included public key to a node name (e.g. ws04.hughesnet.org). They are installed in a web server (or other SSL server) to enable SSL, and provide strong *server to client* authentication. These are similar to the conventional server certs obtained from online Certificate Authorities (CAs).

There are two main differences between server certs generated by PKI_Tool and ones obtained from commercial online CAs:

1. You can create as many server certs as you like with PKI_Tool, with complete control over all aspects of the cert.

2. Issued certs are *private hierarchy*, which means the root and intermediate certs needed to validate the server cert are not normally found in most web servers and browsers, but must be installed in all such software to prevent getting "Untrusted cert" errors upon connect. Once you install the CA certs for such a cert, our certs perform exactly the same function that one from a commercial CA does. Of course, the information in a PKI Tool cert has not been vetted by a trusted third party, but only by the operator of PKI_Tool, if at all. Our server certs are not recommended for servers that may be accessed by the general public, or where Trusted Third Party validation of information in the cert is critical. They work fine in intranets.



Client Certs bind the included public key to a person's name, email address, and or UserID. They are installed in a web browser to provide strong (cryptographic) client to server authentication (in place of username/password authentication). They can also be used for end-to-end secure email with any S/MIME compliant email client, for document signing, or other kinds of authentication in network messaging. As with server certs created by PKI_Tool, the client certs are also private hierarchy, so the relevant root certs must be installed in all relying applications (e.g. web server and all web browsers).

Sixscape

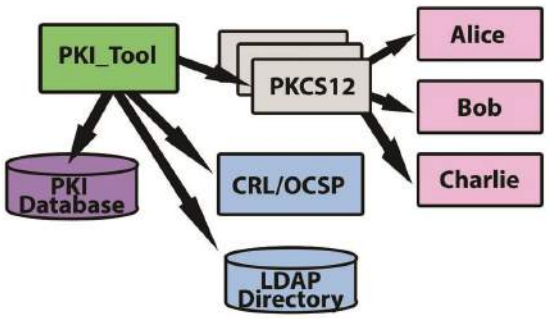
www.sixscape.com

With regards to validation of a client cert applicant's information, the PKI_Tool operator is actually in a much better position to do this than a general online CA. In many cases, such CAs either provide only weakly validated client certs (e.g. that you can receive email at the include email address), or charge a high cost to validate your identity. If you have many applicants that require client certs, the traditional online CA is not really practical, and they do not have access to your HR or customer databases that would provide the identity validation the organization would typically want.

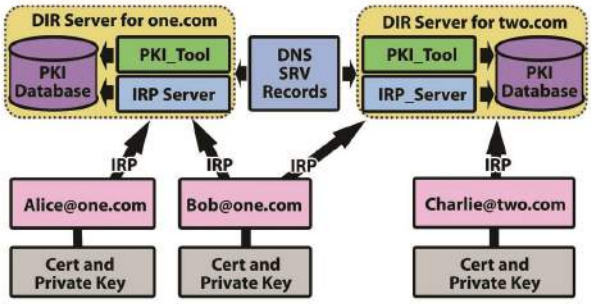
PKI_Tool can create self-signed certs, or entire cert hierarchies. A self-signed cert has no CA certs to chain up to, so each cert must be individually trusted by every user. All of the certs in a given hierarchy can be trusted by installing the CA certs of that hierarchy in the relevant software (e.g. web server and browser). This makes hierarchy certs practical for strong client authentication. PKI Tool makes it very easy to create the root and intermediate cert of a hierarchy, then sign end-entity certs (ones that chain up to the CA certs) using the CA intermediate cert. It is simple to install the CA certs of a PKI Tool hierarchy in a web server, and end users can easily obtain and install these CA certs in their client software (e.g. web browser) via email or by clicking links on a web page.

By itself, PKI_Tool implements a standalone centralized PKI facility. Normally users would be provided with both certificate and private key in a PKCS #12 package or security token. They would not need to create a CSR and private key, or submit the CSR to the PKI through some external means. All key material, including all users' private keys (in encrypted form) is available to the PKI_Tool operator in the PKI_Tool database.

Centralized PKI (Just PKI_Tool)



Distributed PKI (Add IRP Server)



It is possible to deploy the Sixscape Communications Online Certificate Status Protocol (OCSP) Server against the PKI_Tool database to allow OCSP compatible clients to query the revocation status of certificates.

It is possible to install the Sixscape Communications IRP_Server on the same computer, which allows secure access to the PKI_Tool database via the *Identity Registration Protocol* (IRP, port 4604). With IRP, it is also possible for users to create their own CSR and private key, and submit the CSR to the PKI_Tool database, for signing by the PKI Tool operator. They can also create a PKCS #12 container including their certificate and private key, and upload it to the PKI_Tool database, for key material backup and restore (the PKI_Tool operation does not have access to the unencrypted private key in this case). For more details, see the *Domain Identity Registry* (DIR) product brochure. DIR consist of both the PKI_Tool and IRP_Server components, and expands the service into a distributed PKI.

FUNCTIONALITY

- Create CA certs and private keys for a server or client cert hierarchy (with 0, 1 or 2 intermediate certs)
- Create User objects with all information needed to create CSRs or certificates
- Create PKCS #10 Certificate Signing Request (CSR) and private keys for client or server certs
- Sign any PKCS #10 CSR using a hierarchy's lowest CA cert
- Create an X.509 client or server certificate and private key directly (no CSR)
- Create a PKCS #12 secure container from an X.509 certificate and private key, protected with a passphrase (ideal for key material backup/restore or distribution of key material to users)
- Send PKCS #12 container to user via email, or export to a file or USB security token, along with CA certs. This can also be published to a web page for retrieval by users. The PKCS #12 passphrase required to import the key material can be randomly generated and provided securely to each user
- View, Import or Export any PKI object (User, CSR, Certificate, PKCS #12 container)
- Create a Certificate Revocation List periodically for publishing in a website, or provide access to revocation information via an integrated Online Certificate Status Protocol (OCSP) server
- Backup of entire database and PKI objects in XML format
- Retrieval of all (or selected) PKI Objects from any backup file

REQUIREMENTS

PKI_Tool runs on Windows 7 (or later) or Windows Server 2008 (or later). It requires .NET Framework 4.0 Client or better, as well as Microsoft SQL Server 2012 Local database (or SQL Server Express). It does not accept or make network connections. PKI_ Tool is provided in the form of a setup program that installs various components, including the initial empty database.

Supported Standards

PKI_Tool is compliant with the following IETF standards:

- RFC 5280 "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", May 2008

- RFC 2986, "PKCS #10: Certification Request Syntax Specification Version 1.7", November 2000

RFC 5958, "Asymmetric Key Packages", August 2010

SIXSCAPE COMMUNICATIONS PTE LTD

67 Ubi Road 1 Oxley Bizhub #07-10/11
Singapore 408730

Tel | +65 6509 8070 Fax | +65 6509 9667
Email | sales@sixscape.com
URL | www.sixscape.com

Sixscape