Oracle Cryptographic Libraries for SSL

Version 10g, 10.1.0.5

FIPS 140-2 Non-Proprietary
Security Policy

Level 2 Validation
Version 2.0

October 2007
Table of Contents

INTRODUCTION............................................................................................................ 3

PURPOSE....................................................................................................................... 3
REFERENCES .................................................................................................................... 3
DOCUMENT ORGANIZATION ....................................................................................... 3

ORACLE CRYPTOGRAPHIC LIBRARIES FOR SSL................................................... 5

OVERVIEW .................................................................................................................... 5
MODULE INTERFACES .................................................................................................... 6
ROLES AND SERVICES ................................................................................................. 7
   Crypto-Officer Role ..................................................................................................... 8
   User Role .................................................................................................................... 8
   Authentication Mechanisms ...................................................................................... 9

PHYSICAL SECURITY .................................................................................................... 9
OPERATIONAL ENVIRONMENT .................................................................................... 9
CRYPTOGRAPHIC KEY MANAGEMENT ....................................................................... 9
SELF-TESTS .................................................................................................................. 11
   Power-Up Self-tests .................................................................................................. 11
   Conditional Self-tests ............................................................................................... 12
DESIGN ASSURANCE .................................................................................................. 12
MITIGATION OF OTHER ATTACKS ............................................................................. 12

SECURE OPERATION .................................................................................................... 13

INITIAL SETUP ............................................................................................................ 13
CRYPTO-OFFICER GUIDANCE ..................................................................................... 13
USER GUIDANCE ......................................................................................................... 14

ACRONYMS ................................................................................................................... 15

GLOSSARY .................................................................................................................... 16
Introduction

Purpose

This is a non-proprietary Cryptographic Module Security Policy for the Oracle Cryptographic Libraries for SSL from Oracle Corporation (Oracle). The applicable version is 10g, 10.1.0.5. This Security Policy describes how the Oracle Cryptographic Libraries for SSL meets the security requirements of FIPS 140-2 and how to run the module in a secure FIPS 140-2 mode. This policy was prepared as part of the Level 2 FIPS 140-2 validation of the module. FIPS 140-2 (Federal Information Processing Standards Publication 140-2 — Security Requirements for Cryptographic Modules) details the U.S. Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/cryptval/.

Oracle Cryptographic Libraries for SSL is referred to in this document as Cryptographic Library, Software Library, Cryptographic module, software module, or module.

References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The Oracle Corporation website (http://www.oracle.com) contains information on the full line of products from Oracle.
- The CMVP website (http://csrc.nist.gov/cryptval/) contains contact information for answers to technical or sales-related questions for the module.

Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Machine
- Other supporting documentation as additional references

Corsec Security, Inc. under contract to Oracle Corporation produced this Security Policy and the other validation submission documents. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Validation Documentation is proprietary to Oracle Corporation and is releasable only under appropriate non-
disclosure agreements. For access to these documents, please contact Oracle Corporation.
**Oracle Cryptographic Libraries for SSL**

*Overview*

The Cryptographic Library for SSL is a generic module used by Oracle in a variety of its application suites. The module is used to provide support for cryptography, authentication, SSL cryptographic primitives (the module does not implement the SSL protocol), PKCS#11 and certificate management for applications like the Oracle Database (Server & Client) and Oracle Advanced Security Option, Oracle Applications Server and its components including Oracle Identity Management, Oracle Application Server Web Cache and Oracle HTTP Server. It provides a rich set of functionalities and uses PKCS based wallet structures for managing identities and trust points.

The boundary consists of a single shared library (libnnz10.so), which is linked with different Oracle applications, Certicom SSL layer, and data/log files. The libnnz10.so shared library comprises four components given below:

- **NZ library:** This is the interface available to the oracle products for SSL protocol implementation and the cryptographic support required for it. It is a wrapper on top of Certicom, Crypto-C, and CertC toolkits.
- **Crypto-C library:** The Crypto-C library is the cryptographic library used by the module. The algorithms and the RNG implementation are in this library.
- **CertC:** The CertC library is used for certificate processing and management.
- **Certicom SSL Toolkit’s PKCS#12:** This is used for the PKCS#12 structure. For the cryptographic functionality, the Crypto-C library is used.

The logical cryptographic boundary is depicted in the figure below.
The module meets overall level 2 requirements for FIPS 140-2, and Table 1 describes the level achieved by the module in each of the eleven sections of FIPS 140-2 requirements.

<table>
<thead>
<tr>
<th>Section</th>
<th>Section Title</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cryptographic Module Specification</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Cryptographic Module Ports and Interfaces</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Roles, Services, and Authentication</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Finite State Model</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Physical Security</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>Operational Environment</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>Cryptographic Key Management</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>EMI/EMC</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Self-tests</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Design Assurance</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>Mitigation of Other Attacks</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 1 – Security Level per FIPS 140-2 Section

**Module Interfaces**

The Oracle Cryptographic Library for SSL is a multi-chip standalone module that meets overall level 2 FIPS 140-2 requirements. The software cryptographic module is a shared library and is tested for use on a Sun workstation running the Common Criteria (CC) Evaluated Assurance Level (EAL) 4 Solaris Operating
System (OS) version 8 with Admin Suite 3.0.1\(^1\). In addition to the software binary and OS, the module physically consists of a Sun workstation, including the integrated circuits of the motherboard, the central processing unit (CPU), random access memory (RAM), read only memory (ROM), PC case, keyboard, mouse, video interfaces, expansion cards, and other hardware components included in the PC such as hard disk, floppy disk, CD-ROM drive, power supply, and fans.

Logically, the cryptographic boundary of the module is the shared library software running on the operating system. Physically, the cryptographic boundary is defined by the metal enclosure over the Sun workstation (SPARC architecture).

The module’s logical interfaces exist as the public API of the shared library. Additionally, the operating system’s file system is used to interface with the module through data and configuration files. These interfaces mapped to the following FIPS 140-2 logical interfaces as described in the following table.

<table>
<thead>
<tr>
<th>FIPS 140-2 Logical Interface</th>
<th>Module Logical Interface</th>
<th>Physical Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Input Interface</td>
<td>Arguments for a NZ library that specify data to be used or processed by the module.</td>
<td>Keyboard, mouse, CD-ROM, floppy disk, and serial/USB/parallel/network ports</td>
</tr>
<tr>
<td>Data Output Interface</td>
<td>Arguments for a NZ library function that specify where the result of the function is stored. Output data is also stored in a file.</td>
<td>Hard Disk, monitor, and serial/USB/parallel/network ports</td>
</tr>
<tr>
<td>Control Input Interface</td>
<td>Function calls to the NZ library. The Operating system’s file system and configuration files.</td>
<td>Keyboard, CD-ROM, floppy disk, mouse, and serial/USB/parallel/network port</td>
</tr>
<tr>
<td>Status Output Interface</td>
<td>Return values for function calls and log files are the interfaces for status output. The error messages are also logged into files in the OS.</td>
<td>Hard Disk, monitor, and serial/USB/parallel/network ports, error log files</td>
</tr>
<tr>
<td>PC Power Interface</td>
<td>Not Applicable</td>
<td>Power Interface</td>
</tr>
</tbody>
</table>

Table 2 – FIPS 140-2 Logical Interfaces

**Roles and Services**

The module implements cryptographic algorithms including AES, Triple-DES, SHA-1, RSA, PRNG, HMAC, Diffie-Hellman, RC4, RSA-MD5, and HMAC-MD5. The latter three are not available in FIPS mode.

The module supports role-based authentication. There are two roles that operators may assume: a Crypto-Officer role and User role. They are the operators of the Solaris Operating System on which the module is tested. The root user on the Solaris platform and the user that installs the module are the Crypto-Officers; other users of

the module are classified as Users. The module supports multiple concurrent operators.

**Crypto-Officer Role**

The Crypto-Officer role is responsible for initializing and maintaining the module. Descriptions of the services available to the Crypto-Officer role are provided in the table below.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Input</th>
<th>Output</th>
<th>CSP</th>
<th>Type of Access to CSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>Installing the software module</td>
<td>Command</td>
<td>Result of installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uninstall</td>
<td>Uninstall the cryptographic Library</td>
<td>Command</td>
<td>Module uninstalled</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3 – Crypto-Officer Services**

**User Role**

The User role is able to access the general services provided by the module, through the NZ API provided by the module. Descriptions of the services available to the User role are provided in the table below.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Input</th>
<th>Output</th>
<th>Accessed CSPs</th>
<th>Access Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NZ layer operation</td>
<td>Start, close a NZ session</td>
<td>API call</td>
<td>Status output</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation on Oracle wallet</td>
<td>Create, delete, open, close, and copy a wallet</td>
<td>API calls with wallet information</td>
<td>Status output, result of command</td>
<td>RSA key pair</td>
<td>Read/Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wallet Password</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Manipulate information stored in a wallet</td>
<td>Store or retrieve RSA key pairs, CA or any other information to/from a wallet</td>
<td>API calls with wallet information</td>
<td>Status output, result of command</td>
<td>RSA key pair</td>
<td>Read/Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CA, Trustpoint</td>
<td>Read/Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wallet Password</td>
<td>Read/Write</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Session key</td>
<td>Read</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Symmetric key</td>
<td>Read</td>
</tr>
<tr>
<td>Choose Cipher Suite</td>
<td>User may choose a cipher suite for the SSL layer from given options.</td>
<td>API calls</td>
<td>Status output</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cipher operation</td>
<td>Encrypt/decrypt of data</td>
<td>API calls</td>
<td>Status output, result of command</td>
<td>Symmetric key</td>
<td>Read</td>
</tr>
<tr>
<td>Sign/verify</td>
<td>Sign and verify of data</td>
<td>API calls with RSA key pair</td>
<td>Status output, result of command</td>
<td>RSA key pair</td>
<td>Read</td>
</tr>
<tr>
<td>Key pair operation</td>
<td>Generate, delete, copy RSA key pair; set or get info</td>
<td>API calls with RSA key pair</td>
<td>Status output, result of command</td>
<td>RSA key pair</td>
<td>Read/Write</td>
</tr>
</tbody>
</table>
Service | Description | Input | Output | Accessed CSPs | Access Type
---|---|---|---|---|---
Key agreement | Diffie-Hellman key agreement protocol in different phases | API calls, DH key pairs, Shared Secret | Status output, DH key agreement | DH key pairs | Read/Write

**Table 4 - User Services**

*Authentication Mechanisms*

The module supports a username/password based authentication mechanism. Each valid username has an associated role, and the username can only be used to gain access to the module with the associated password.

The implementation is the one that is used by the CC EAL 4 certified Solaris operating system of the module. The minimum length of the password is 8 and the module requires the password to be alphanumeric. Assuming only 36 characters (A-Z, a-z, and 0-9) with repetition, the chance of a random attempt falsely succeeding is 1 in 2821109907456. A typical workstation will process 20000 requests per minute. So the probability of a successful random attempt will be around 20000/2821109907456 which is less than 1 in 100000.

*Physical Security*

Physical security requirements do not apply to this module. The Oracle software library is a software module and does not implement any physical security mechanisms.

While purely a software module, the FIPS 140-2 tested platform must be a Solaris PC (SPARC architecture) that has been tested for and meets applicable FCC EMI and EMC requirements for business use as defined by 47 Code of Federal Regulations, Part15, Subpart B.

*Operational Environment*

The Oracle Cryptographic Library is a software cryptographic module running on the CC EAL 4 certified Sun Solaris operating system version 8 with Admin Suite 3.0.1\(^2\), and the module’s software is entirely encapsulated by the cryptographic boundary shown in Figure 1.

*Cryptographic Key Management*

The module implements the following FIPS-approved algorithms using the underlying Crypto-C 6.0.2 library:

- AES – CBC mode (certificate #608)
- Triple-DES – CBC mode (certificate #573)
- SHA-1 – Byte oriented (certificate #657)
- RSA signature generation/verification (PKCS#1) – (certificate #281)
- PRNG – FIPS 186-2 Appendix 3.1 (certificate #347)
- HMAC-SHA-1 (certificate #314).

Additionally, the module implements the following non-FIPS-approved algorithms that can be used in the FIPS mode:

- RSA key wrapping, 1024, 2048, and 4096 bits, providing 80, 112, and 150 bits of security, respectively
- Diffie-Hellman (Diffie-Hellman provides 80-bits of security).

And the following non-FIPS-approved algorithms that can not be used in FIPS mode:

- RC4
- RSA-MD5
- HMAC-MD5.

The module supports the following critical security parameters:

<table>
<thead>
<tr>
<th>Key or CSP</th>
<th>Key type</th>
<th>Generation</th>
<th>Storage</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA private key</td>
<td>1024, 2048 or 4096 bit keys</td>
<td>Generated internally and externally</td>
<td>Encrypted or plaintext in hard drive</td>
<td>S/MIME sign and encrypt and key agreement</td>
</tr>
<tr>
<td>RSA public key</td>
<td>1024, 2048 or 4096 bit keys</td>
<td>Generated internally and externally</td>
<td>Encrypted or plaintext in hard drive</td>
<td>S/MIME verify and decrypt and key agreement</td>
</tr>
<tr>
<td>Diffie-Hellman private key</td>
<td>1024 bits Diffie-Hellman private key</td>
<td>Generated internally and externally</td>
<td>Plaintext in volatile memory</td>
<td>Used for key in key establishment with external application</td>
</tr>
<tr>
<td>Diffie-Hellman public key</td>
<td>1024 bits Diffie-Hellman public key</td>
<td>Generated internally and externally</td>
<td>Plaintext in volatile memory</td>
<td>Used for key in key establishment with external application</td>
</tr>
<tr>
<td>Shared Secret</td>
<td>512, 768, 1024, 2048 or 4096 bit</td>
<td>Generated internally</td>
<td>Plaintext in volatile memory</td>
<td>Used in Certicom SSL layer (out of Cryptographic Boundary)</td>
</tr>
<tr>
<td>Session Key</td>
<td>64 bits</td>
<td>Generated externally</td>
<td>Plaintext in volatile memory</td>
<td>Used in Certicom SSL layer (out of Cryptographic Boundary)</td>
</tr>
<tr>
<td>TDES CBC</td>
<td>192 bits</td>
<td>Generated internally</td>
<td>Plaintext in volatile memory</td>
<td>Used in external application</td>
</tr>
<tr>
<td>AES CBC</td>
<td>128, 256 bits</td>
<td>Generated internally</td>
<td>Plaintext in volatile memory</td>
<td>Used in external application</td>
</tr>
</tbody>
</table>
### Table 5 – Listing of Key and Critical Security Parameters

<table>
<thead>
<tr>
<th>Software Integrity test key</th>
<th>HMAC key</th>
<th>Hard Coded</th>
<th>Plaintext in hard drive and volatile memory</th>
<th>Software integrity test using HMAC-SHA-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Password for Oracle wallet</td>
<td>Entered electronically</td>
<td>Plaintext in file system</td>
<td>Encrypts/decrypts Oracle wallet</td>
</tr>
<tr>
<td>User password</td>
<td>Sun Solaris OS password</td>
<td>Entered electronically</td>
<td>Sun workstation registry in plaintext</td>
<td>Authenticate users to OS</td>
</tr>
<tr>
<td>PRNG seed</td>
<td>256 bits</td>
<td>Generated internally</td>
<td>Plaintext in volatile memory</td>
<td>Seeds PRNG</td>
</tr>
</tbody>
</table>

The cryptographic module generates a RSA key pair using PRNG. The Oracle Cryptographic Library securely administers all of its CSPs, which include a RSA Public/Private key pair, Diffie-Hellman key pairs, ephemeral session and Shared Secret keys, Symmetric (TDES and AES) keys, HMAC, and the PRNG seed. Diffie-Hellman key pairs enter or exit the module in plaintext; they need not be encrypted as their strength relies on the underlying protocol. Session keys also enter or are output from the module in plaintext.

RSA key pairs cross the cryptographic boundary in plaintext. RSA key pairs can be zeroized using proper function calls. Diffie-Hellman key pairs are used for key establishment protocol with external application and zeroized after a session is established. Shared Secret is being generated internally and output in plaintext. The CSP is zeroized after use or by rebooting power cycle. Session Key is generated and entered in the module in clear. It is being zeroized after use or by rebooting power cycle. Symmetric keys (TDES and AES) are zeroized when the session with external application is over. The Integrity Test Key is a hard-coded key that is included with the module. The HMAC key is zeroized when the module is uninstalled. PRNG seed is overwritten when another block is generated and can be zeroized by rebooting.

**Self-Tests**

In order to prevent any secure data from being released, it is important to test the cryptographic components of a security module to ensure all components are functioning correctly. To confirm correct functionality, the software library performs the following self-tests:

**Power-Up Self-tests**

- Software integrity check: Verifying the integrity of the software binaries of the module with HMAC SHA-1.
- Triple-DES (CBC) KAT: Verifying the correct operation of the Triple-DES algorithm implementation.
• AES (CBC) KAT: Verifying the correct operation of the AES algorithm implementation.

• RSA encrypt/decrypt pair-wise consistency check: Verifying the correct operation of the RSA implementation.

• SHA-1 KAT: Verifying the correct operation of the SHA-1 algorithm implementation.

• FIPS 186-2 RNG KAT: Verifying the correct operation of the FIPS 186-2 RNG implementation.

Conditional Self-tests

• RSA encrypt/decrypt pair-wise consistency check: Verifying that a newly generated or stored RSA key pair works properly.

• RSA sign/verify pair-wise consistency check: Verifying the correct operation of the RSA implementation.

• FIPS 186-2 Continuous RNG: Verifying the RNG has not failed by producing a constant value.

• Continuous RNG: Verifying the non-approved RNG (used for seeding FIPS approved PRNG) does not generate the same value through repeated attempts.

The module provides any cryptographic functionality only after all the power-up self-tests are successful. Results of self-tests are logged in a trace file for the Crypto-Officer’s review. If any of the self-tests fail, the module enters an error state that can be cleared by cycling the computer’s power.

Design Assurance

Source code and associated documentation files are managed and recorded using an Oracle-developed content management system called the “Advanced Development Environment,” or ADE. The ADE contains a database of files with full modification history. For every modification, the ADE assigns a new version number and records how, when and by who the file has been modified. In addition, a short comment is attached to the modified file by the developer.

Mitigation of Other Attacks

In a FIPS mode of operation, the module does not claim to mitigate any attacks.
SECURE OPERATION

The Oracle Cryptographic Library for SSL meets Level 2 requirements for FIPS 140-2. The sections below describe how to place and keep the module in FIPS-approved mode of operation.

Initial Setup

The module is to be installed on a Sun Solaris operating system version 8 with Admin Suite 3.0.1. The operating system has to be installed and configured to meet the CC EAL 4 evaluated installation and configuration procedures. The secured configuration process for the OS can be found in the Security Target document. The module is delivered in a secure manner. The Crypto-Officer buys the CD for the module or the application it is linked to from Oracle. Oracle ships the module in a sealed box with Oracle’s seal across the box. Upon receipt of the module, the Crypto-Officer must examine the package for evidence of tampering. Tamper-evidence includes tears, scratches, and other irregularities in the packaging.

’sqlnet.ora’ file is created during the network configuration tasks for authentication or encryption. Configuration parameters for network security are in the sqlnet.ora file and are different for each of the client and server processes. Only the Crypto-Officer has ‘write’ access to sqlnet.ora. The default location for sqlnet.ora is $ORACLE_HOME/network/admin. However, the location can be specified by configuring the environment variable TNS_ADMIN.

Crypto-Officer Guidance

The Crypto-Officer is required to configure the module to run in a FIPS mode of operation. To do so, the Crypto-Officer sets the variable ‘SQLNET.SSLFIPS_140’ to TRUE in the ‘sqlnet.ora’ file. The default setting of the FIPS_140 parameter is FALSE. The crypto officer must set this parameter to TRUE in the client and server sqlnet.ora configuration files in order to be FIPS 140-2 compliant.

Configuration parameters defined in the sqlnet.ora file can enable or disable net tracing. Enabling tracing allows the module to log status of the self tests. Status messages of self tests would be logged in the tracing file indicated in the configuration parameter by the Crypto-Officer. Below is an example of how to set tracing file.

```
trace_level_owm=16
trace_directory=/private/oracle/server
trace_file_server=svr.trc
```


© Copyright 2007 Oracle Corporation

This document may be freely reproduced and distributed whole and intact including this Copyright Notice.
**User Guidance**

The User accesses the module’s functionality as a client. Although the User does not have any ability to modify the configuration of the module, care should be taken not to provide authentication information (Wallet Password) to other parties. The User is also responsible for selecting SSL cipher suites that use only FIPS approved cryptographic algorithms.
ACRONYMS

AH Authentication Header
ANSI American National Standards Institute
API Application Programming Interface
CC Common Criteria
CLI Command Line Interface
CMVP Cryptographic Module Validation Program
CRC Cyclic Redundancy Check
CSE Communications Security Establishment
CSP Critical Security Parameter
DSA Digital Signature Algorithm
EAL Evaluated Assurance Level
EDC Error Detection Code
EMC Electromagnetic Compatibility
EMI Electromagnetic Interference
ESP Encapsulating Security Payload
FCC Federal Communication Commission
FIPS Federal Information Processing Standard
HMAC (Keyed-) Hash MAC
IKE Internet Key Exchange
IP Internet Protocol
IPSec IP Security
KAT Known Answer Test
LED Light Emitting Diode
MAC Message Authentication Code
NIST National Institute of Standards and Technology
NVLAP National Voluntary Laboratory Accreditation Program
OS Operating System
OWM Oracle Wallet Manager
PC Personal Computer
RAM Random Access Memory
RIP Routing Information Protocol
RNG Random Number Generator
RSA Rivest Shamir and Adleman
SA Security Association
SHA Secure Hash Algorithm
SNMP Simple Network Management Protocol
SP Secure Platform
SSH Secure SHell
SSL Secure Socket Layer
TCP Transmission Control Protocol
TLS Transport Layer Security
VSS Visual Source Safe
VPN Virtual Private Network
GLOSSARY

PKCS #12
PKCS #12 is an industry standard for storing and transferring personal authentication credentials.

Trusted Certificate
A trusted certificate, sometimes called a root key certificate, is a third party identity that is qualified with a level of trust. The trusted certificate is used when an identity is being validated as the entity it claims to be. Typically, the certificate authorities you trust are called trusted certificates. If there are several levels of trusted certificates, a trusted certificate at a lower level in the certificate chain does not need to have all its higher level certificates reverified.

Trust Point
See trusted certificate.

Wallet
A wallet is a data structure used to store and manage security credentials for an individual entity. It implements the storage and retrieval of credentials for use with various cryptographic services. A Wallet Resource Locator (WRL) provides all the necessary information to locate the wallet.

Wallet Resource Locator
A wallet resource locator (WRL) provides all necessary information to locate a wallet. It is a path to an operating system directory that contains a wallet.

X.509
Public keys can be formed in various data formats. The X.509 v3 format is one such popular format.