

OpenSSL A.00.09.07m.020,
A.00.09.07m.021, and
A.00.09.08g.019 Release Notes
HP-UX: 11i v1, 11i v2, 11i v3



© Copyright 2008 Hewlett-Packard Development Company, L.P.

Legal Notices

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

UNIX is a registered trademark of The Open Group. Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Intel and Itanium are trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

1 OpenSSL A.00.09.07m.020, A.00.09.07m.021, and A.00.09.08g.019

This document contains the most recent product information for OpenSSL A.00.09.07m.020, A.00.09.07m.021, and A.00.09.08g.019 supported on HP-UX 11i v1, HP-UX 11i v2, and HP-UX 11i v3. Use this document for the following information:

- OpenSSL Features
- Installing OpenSSL
- Using the OpenSSL command-line Tool
- Frequently Asked Questions (FAQs)

Announcement

This version of OpenSSL is based on the open source OpenSSL 0.9.7m and 0.9.8g products. This bundle contains the following:

- OpenSSL A.00.09.08g in the `/opt/openssl/0.9.8` directory
- OpenSSL A.00.09.07m in the `/opt/openssl/0.9.7` directory
- FIPS Capable OpenSSL (based on 0.9.7m and linked against FIPS 140-2 module) in the `/opt/openssl/fips/0.9.7` directory

The default version of OpenSSL that is enabled on HP-UX 11i v1 and HP-UX 11i v2 is OpenSSL A.00.09.07m. The default version of OpenSSL on HP-UX 11i v3 is OpenSSL A.00.09.08g. Use the `/opt/openssl/switchversion.sh` script to switch between OpenSSL A.00.09.07m and OpenSSL A.00.09.08g. You can also use this script to swap the `openssl.cnf` file depending on the version of OpenSSL. However, this is an optional step.

OpenSSL A.00.09.07m and A.00.09.08g offer a general-purpose cryptography library and implementation of the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. OpenSSL A.00.09.07m and A.00.09.08g is fully tested and supported on the HP-UX operating system.

OpenSSL A.00.09.07m and A.00.09.08g is an upgrade for HP-UX OpenSSL Versions A.00.09.07l, and A.00.09.08d.

OpenSSL A.00.09.07m and A.00.09.08g is available as a Web upgrade. The software bits are available at: <http://www.software.hp.com>.

Install the appropriate OpenSSL version based on your operating system. Table 1-1 lists the OpenSSL releases available for HP-UX 11i v1, HP-UX 11i v2, and HP-UX 11i v3.

Table 1-1 Default OpenSSL Versions for HP-UX 11i v1 HP-UX 11i v2, and HP-UX 11i v3

Supported Operating Systems	OpenSSL Version
HP-UX 11i v1	OpenSSL A.00.09.07m.020
HP-UX 11i v2	OpenSSL A.00.09.07m.021
HP-UX 11i v3	OpenSSL A.00.09.08g.019

OpenSSL A.00.09.07m has the following options:

```
./Configure threads zlib shared no-rc5 no-idea no-krb5  
no-mdc2 -openssldir=/opt/openssl hpux-cc
```

OpenSSL A.00.09.08g has the following options:

```
./Configure threads zlib shared no-rc5 no-idea no-krb5 -openssldir=/opt/openssl  
hpux-cc
```

FIPS Capable OpenSSL is built with the following options:

```
./config fips for FIPS module  
./Configure fips zlib threads no-rc5 no-idea no-krb5 no-mdc2  
--openssldir=/opt/openssl for FIPS Compatible OpenSSL
```

Where:

threads	Creates a library suitable for multi threaded applications.
zlib	Provides support for zlib compression.
shared	Builds shared libraries.
no-rc5	Builds OpenSSL without the Rivest encryption Cipher 5 (RC5) cipher algorithm.
no-idea	Builds OpenSSL without the International Data Encryption Algorithm (IDEA) cipher.
no-krb5	Directs OpenSSL not to compile in any Kerberos 5 (KRB5) library or code.
no-mdc2	(MDC2) library or code.
--prefix	Specifies the prefix for the OpenSSL include, lib, and bin directories.

OpenSSL Versions A.00.09.07m and A.00.09.08g use different cryptographic algorithms to perform operations, such as authenticating the server and client to each other, transmitting certificates, and establishing session keys.

New in OpenSSL A.00.09.07m and A.00.09.08g

Federal Information Processing Standard (FIPS) 140-2 OpenSSL is now added to the OpenSSL product. For more information about FIPS 140-2, see the following web address:

http://www.oss-institute.org/index.php?option=com_content&view=article&id=84&Itemid=123



IMPORTANT: The FIPS code is certified only if it is identical with the source code released by the Open Source Software Institute (OSSI) organization on the OpenSSL website. In the event of a security vulnerability, HP cannot modify the source code because a modification of the source code can invalidate the certification.

If a vulnerability is found in the FIPS code, HP will wait until the OSSI organization releases a new FIPS 140-2 certified FIPS module before updating the HP OpenSSL product with the new FIPS code.

This release of OpenSSL also contains some minor enhancements included in OpenSSL A.00.09.07m and A.00.09.08g. For more information, see the OpenSSL Changelog at: <http://www.openssl.org/news/changelog.html>

What is in OpenSSL A.00.09.07m and A.00.09.08g

OpenSSL A.00.09.07m and A.00.09.08g support the following security features:

- Ciphers
- Message Digest
- Public key encryption
- Certificates
- Encoding

The following sections discuss each of the security features in detail.

Ciphers

A cipher algorithm is a mechanism used to encrypt or decrypt a message. OpenSSL A.00.09.07m and A.00.09.08g support the following ciphers:

- Blowfish
- Carlisle Adams and Stafford Tavares (CAST)
- Advanced Encryption Standard (AES)
- Data Encryption Standard (DES)



WARNING! DES has been cracked (data encoded by DES has been decoded by a third party). HP recommends that you use DES only when you are required to do so for compatibility reasons or because of legal restrictions.

- Triple Data Encryption Standard (3DES)
- Data Encryption Standard Extension (DESX)
- Rivest Cipher 2 (RC2)
- Rivest Cipher 4 (RC4)

Message Digest

A message digest is a piece of data that can be used to verify that the contents of the message has not been altered during transit. When a message is sent over a network, the sender computes a message digest by performing a one-way hash function using a secret key known only to the sender and recipient. The recipient also computes the message digest by performing the same one-way hash function using the secret key. If the two message digests are identical, the recipient can be sure that the message had not been modified during transit.

OpenSSL A.00.09.07m and A.00.09.08g support the following message digest algorithms:

- Hashed Message Authentication Code (HMAC)
- Message Digest 2 (MD2) algorithm
- Message Digest 4 (MD4) algorithm
- Message Digest 5 (MD5) algorithm
- RACE Integrity Primitives Evaluation Message Digest (RIPEMD) algorithm
- Secure Hash Algorithm (SHA)
- Secure Hash Algorithm 1 (SHA1)

Public Key Encryption

Public-key encryption is an asymmetric encryption method that uses a public key and a private key to encrypt and decrypt messages.

OpenSSL A.00.09.07m and A.00.09.08g support the following public key encryption methods:

- Rivest, Shamir, and Adleman (RSA) algorithm
- Digital Signature Algorithm (DSA)
- Diffie-Hellman (DH) algorithm

Certificates

A digital certificate is a file that uniquely identifies users and resources over a network.

OpenSSL A.00.09.07m and A.00.09.08g support the following digital certificates:

- X.509
- X.509 Version 3
- Certificate Revocation List (CRL)

Encoding

Before a message is sent over a network, the message is encoded such that the receiver can understand the message. OpenSSL A.00.09.07m and A.00.09.08g support the following file formats for encoding keys, certificates, and digitally signed files:

- Distinguished Encoding Rules (DER) – Stores Abstract Syntax Notation One (ASN.1) structures containing keys and certificates.
- Privacy Enhanced Mail (PEM) – Stores keys, certificates, and encrypted files.
- Public-Key Cryptography Standard 7 (PKCS#7) – Stores digitally signed files.
- Public-Key Cryptography Standard 8 (PKCS#8) – Stores private keys.
- Public-Key Cryptography Standard 12 (PKCS#12) – Stores keys and certificates in browsers.

What is in OpenSSL A.00.09.08g

OpenSSL A.00.09.08g supports all the security features that are available in OpenSSL A.00.09.07m. In addition, OpenSSL A.00.09.08g also supports the following public-key encryptions:

- Elliptic Curve Crypto (ECC)
- Elliptic Curve Diffie-Hellman (ECDH)
- Elliptic Curve Digital Signature Algorithm (ECDSA)

OpenSSL A.00.09.08g also provides library support to the following hardware ENGINES:

- 4758cca
- aep
- atalla
- chil
- cswift
- gmp
- nuron
- sureware
- ubsec

OpenSSL Components

OpenSSL A.00.09.07m and A.00.09.08g contain the following components:

- OpenSSL libraries
- The `openssl` command-line tool
- Strong Random Number Generator for HP-UX 11i v1
- Automatically generated self-signed host certificate

The following sections discuss these components in detail.

OpenSSL Libraries

OpenSSL A.00.09.07m and A.00.09.08g contain two libraries: `libcrypto` and `libssl`. The `libcrypto` library contains all the cryptographic functions used for creating and managing ciphers, digests, certificates, public key encryption, and encoding. The `libssl` library contains all the functions used for managing secure connections between SSL-enabled clients and the corresponding SSL-enabled servers.

OpenSSL A.00.09.07m and A.00.09.08g provides 32-bit and 64-bit libraries for static and shared versions of both the libraries.

A number of symbolic links are created when OpenSSL A.00.09.07m and A.00.09.08g is installed on the system. These symbolic links are listed in the following tables:

Table 1-2 OpenSSL A.00.09.07m PA-RISC Libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/0.9.7/lib/libssl.0.9.7m.a	<ul style="list-style-type: none"> • /usr/lib/libssl.a * • /opt/openssl/lib/libssl.a * • /opt/openssl/0.9.7/lib/libssl.a • /opt/openssl/0.9.8/lib/libssl.0.9.7m.a
	/opt/openssl/0.9.7/lib/libcrypto.0.9.7m.a	<ul style="list-style-type: none"> • /usr/lib/libcrypto.a * • /opt/openssl/lib/libcrypto.a * • /opt/openssl/0.9.7/lib/libcrypto.a • /opt/openssl/0.9.8/lib/libcrypto.0.9.7m.a
32-bit shared	/opt/openssl/0.9.7/lib/libssl.sl.0	<ul style="list-style-type: none"> • /usr/lib/libssl.sl * • /usr/lib/libssl.sl.0 • /opt/openssl/lib/libssl.sl * • /opt/openssl/lib/libssl.sl.0 • /opt/openssl/0.9.7/lib/libssl.sl • /opt/openssl/0.9.8/lib/libssl.sl.0
	/opt/openssl/0.9.7/lib/libcrypto.sl.0	<ul style="list-style-type: none"> • /usr/lib/libcrypto.sl * • /usr/lib/libcrypto.sl.0 • /opt/openssl/lib/libcrypto.sl * • /opt/openssl/lib/libcrypto.sl.0 • /opt/openssl/0.9.7/lib/libcrypto.sl • /opt/openssl/0.9.8/lib/libcrypto.sl.0
64-bit static	/opt/openssl/0.9.7/lib/pa20_64/libssl.0.9.7m.a	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libssl.a * • /opt/openssl/lib/pa20_64/libssl.a * • /opt/openssl/0.9.7/lib/pa20_64/libssl.a • /opt/openssl/0.9.8/lib/pa20_64/libssl.0.9.7m.a
	/opt/openssl/0.9.7/lib/pa20_64/libcrypto.0.9.7m.a	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libcrypto.a * • /opt/openssl/lib/pa20_64/libcrypto.a * • /opt/openssl/0.9.7/lib/pa20_64/libcrypto.a • /opt/openssl/0.9.8/lib/pa20_64/libcrypto.0.9.7m.a

Table 1-2 OpenSSL A.00.09.07m PA-RISC Libraries (continued)

Library	Library Name/Location	Symbolic Link
64-bit shared	/opt/openssl/0.9.7/lib/pa20_64/libssl.sl.0	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libssl.sl * • /usr/lib/pa20_64/libssl.sl.0 • /opt/openssl/lib/pa20_64/libssl.sl * • /opt/openssl/lib/pa20_64/libssl.sl.0 • /opt/openssl/0.9.7/lib/pa20_64/libssl.sl • /opt/openssl/0.9.8/lib/pa20_64/libssl.sl.0
	/opt/openssl/0.9.7/lib/pa20_64/libcrypto.sl.0	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libcrypto.sl * • /usr/lib/pa20_64/libcrypto.sl.0 • /opt/openssl/lib/pa20_64/libcrypto.sl * • /opt/openssl/lib/pa20_64/libcrypto.sl.0 • /opt/openssl/0.9.7/lib/pa20_64/libcrypto.sl • /opt/openssl/0.9.8/lib/pa20_64/libcrypto.sl.0



NOTE: Symbolic links marked * are applicable only if the default version is OpenSSL A.00.09.07m.

Table 1-3 OpenSSL A.00.09.07m Intel® Itanium® Libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/0.9.7/lib/hpux32/libssl.0.9.7m.a	<ul style="list-style-type: none">• /usr/lib/hpux32/libssl.a *• /opt/openssl/lib/hpux32/libssl.a *• /opt/openssl/0.9.7/lib/hpux32/libssl.a• /opt/openssl/0.9.8/lib/hpux32/libssl.0.9.7m.a
	/opt/openssl/0.9.7/lib/hpux32/libcrypto.0.9.7m.a	<ul style="list-style-type: none">• /usr/lib/hpux32/libcrypto.a *• /opt/openssl/lib/hpux32/libcrypto.a *• /opt/openssl/0.9.7/lib/hpux32/libcrypto.a• /opt/openssl/0.9.8/lib/hpux32/libcrypto.0.9.7m.a
32-bit shared	/opt/openssl/0.9.7/lib/libssl.sl.0	<ul style="list-style-type: none">• /usr/lib/hpux32/libssl.so *• /usr/lib/hpux32/libssl.so.0• /opt/openssl/lib/hpux32/libssl.so *• /opt/openssl/lib/hpux32/libssl.so.0• /opt/openssl/0.9.7/lib/hpux32/libssl.so• /opt/openssl/0.9.8/lib/hpux32/libssl.so.0
	/opt/openssl/lib/hpux32/libcrypto.so.0	<ul style="list-style-type: none">• /usr/lib/hpux32/libcrypto.so *• /usr/lib/hpux32/libcrypto.so.0• /opt/openssl/lib/hpux32/libcrypto.so *• /opt/openssl/lib/hpux32/libcrypto.so.0• /opt/openssl/0.9.7/lib/hpux32/libcrypto.so• /opt/openssl/0.9.8/lib/hpux32/libcrypto.so.0
64-bit static	/opt/openssl/0.9.7/lib/hpux64/libssl.0.9.7m.a	<ul style="list-style-type: none">• /usr/lib/hpux64/libssl.a *• /opt/openssl/lib/hpux64/libssl.a *• /opt/openssl/0.9.7/lib/hpux64/libssl.a• /opt/openssl/0.9.8/lib/hpux64/libssl.0.9.7m.a
	/opt/openssl/0.9.7/lib/hpux64/libcrypto.0.9.7m.a	<ul style="list-style-type: none">• /usr/lib/hpux64/libcrypto.a *• /opt/openssl/lib/hpux64/libcrypto.a *• /opt/openssl/0.9.7/lib/hpux64/libcrypto.a• /opt/openssl/0.9.8/lib/hpux64/libcrypto.0.9.7m.a

Table 1-3 OpenSSL A.00.09.07m Intel® Itanium® Libraries (continued)

Library	Library Name/Location	Symbolic Link
64-bit shared	/opt/openssl/0.9.7/lib/hpux64/libssl.so.0	<ul style="list-style-type: none"> • /usr/lib/hpux64/libssl.so * • /usr/lib/hpux64/libssl.so.0 • /opt/openssl/lib/hpux64/libssl.so * • /opt/openssl/lib/hpux64/libssl.so.0 • /opt/openssl/0.9.7/lib/hpux64/libssl.so • /opt/openssl/0.9.8/lib/hpux64/libssl.so.0
	/opt/openssl/0.9.7/lib/hpux64/libcrypto.so.0	<ul style="list-style-type: none"> • /usr/lib/hpux64/libcrypto.so * • /usr/lib/hpux64/libcrypto.so.0 • /opt/openssl/lib/hpux64/libcrypto.so * • /opt/openssl/lib/hpux64/libcrypto.so.0 • /opt/openssl/0.9.7/lib/hpux64/libcrypto.so • /opt/openssl/0.9.8/lib/hpux64/libcrypto.so.0



NOTE: Symbolic links marked * are applicable only if the default version is OpenSSL A.00.09.07m.

Table 1-4 OpenSSL A.00.09.08g PA-RISC Libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/0.9.8/lib/libssl.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/libssl.a * • /opt/openssl/lib/libssl.a * • /opt/openssl/0.9.8/lib/libssl.a • /opt/openssl/0.9.7/lib/libssl.0.9.8g.a
	/opt/openssl/0.9.8/lib/libcrypto.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/libcrypto.a * • /opt/openssl/lib/libcrypto.a * • /opt/openssl/0.9.8/lib/libcrypto.a • /opt/openssl/0.9.7/lib/libcrypto.0.9.8g.a
32-bit shared	/opt/openssl/0.9.8/lib/libssl.sl.1	<ul style="list-style-type: none"> • /usr/lib/libssl.sl * • /usr/lib/libssl.sl.1 • /opt/openssl/lib/libssl.sl * • /opt/openssl/lib/libssl.sl.1 • /opt/openssl/0.9.8/lib/libssl.sl • /opt/openssl/0.9.7/lib/libssl.sl.1
	/opt/openssl/0.9.8/lib/libcrypto.sl.1	<ul style="list-style-type: none"> • /usr/lib/libcrypto.sl * • /usr/lib/libcrypto.sl.1 • /opt/openssl/lib/libcrypto.sl * • /opt/openssl/lib/libcrypto.sl.1 • /opt/openssl/0.9.8/lib/libcrypto.sl • /opt/openssl/0.9.7/lib/libcrypto.sl.1
64-bit static	/opt/openssl/0.9.8/lib/pa20_64/libssl.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libssl.a * • /opt/openssl/lib/pa20_64/libssl.a * • /opt/openssl/0.9.8/lib/pa20_64/libssl.a • /opt/openssl/0.9.7/lib/pa20_64/libssl.0.9.8g.a
	/opt/openssl/0.9.8/lib/pa20_64/libcrypto.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libcrypto.a * • /opt/openssl/lib/pa20_64/libcrypto.a * • /opt/openssl/0.9.8/lib/pa20_64/libcrypto.a • /opt/openssl/0.9.7/lib/pa20_64/libcrypto.0.9.8g.a

Table 1-4 OpenSSL A.00.09.08g PA-RISC Libraries (continued)

Library	Library Name/Location	Symbolic Link
64-bit shared	/opt/openssl/0.9.8/lib/pa20_64/libssl.sl.1	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libssl.sl * • /usr/lib/pa20_64/libssl.sl.1 • /opt/openssl/lib/pa20_64/libssl.sl * • /opt/openssl/lib/pa20_64/libssl.sl.1 • /opt/openssl/0.9.8/lib/pa20_64/libssl.sl • /opt/openssl/0.9.7/lib/pa20_64/libssl.sl.1
	/opt/openssl/0.9.8/lib/pa20_64/libcrypto.sl.1	<ul style="list-style-type: none"> • /usr/lib/pa20_64/libcrypto.sl * • /usr/lib/pa20_64/libcrypto.sl.1 • /opt/openssl/lib/pa20_64/libcrypto.sl * • /opt/openssl/lib/pa20_64/libcrypto.sl.1 • /opt/o> • /opt/openssl/0.9.7/lib/pa20_64/libcrypto.sl.1



NOTE: Symbolic links marked * are applicable only if the default version is OpenSSL A.00.09.08g.

Table 1-5 OpenSSL A.00.09.08d Intel Itanium Libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/0.9.8/lib/hpux32/libssl.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/hpux32/libssl.a * • /opt/openssl/lib/hpux32/libssl.a * • /opt/openssl/0.9.8/lib/hpux32/libssl.a • /opt/openssl/0.9.7/lib/hpux32/libssl.0.9.8g.a
	/opt/openssl/0.9.8/lib/hpux32/libcrypto.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/hpux32/libcrypto.a * • /opt/openssl/lib/hpux32/libcrypto.a * • /opt/openssl/0.9.8/lib/hpux32/libcrypto.a • /opt/openssl/0.9.7/lib/hpux32/libcrypto.0.9.8g.a
32-bit shared	/opt/openssl/0.9.8/lib/hpux32/libssl.so.1	<ul style="list-style-type: none"> • /usr/lib/hpux32/libssl.so * • /usr/lib/hpux32/libssl.so.1 • /opt/openssl/lib/hpux32/libssl.so * • /opt/openssl/lib/hpux32/libssl.so.1 • /opt/openssl/0.9.8/lib/hpux32/libssl.so • /opt/openssl/0.9.7/lib/hpux32/libssl.so.1
	/opt/openssl/0.9.8/lib/hpux32/libcrypto.so.1	<ul style="list-style-type: none"> • /usr/lib/hpux32/libcrypto.so * • /usr/lib/hpux32/libcrypto.so.1 • /opt/openssl/lib/hpux32/libcrypto.so * • /opt/openssl/> • /opt/openssl/0.9.8/lib/hpux32/libcrypto.so • /opt/openssl/0.9.7/lib/hpux32/libcrypto.so.1
64-bit static	/opt/openssl/0.9.8/lib/hpux64/libssl.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/hpux64/libssl.a * • /opt/openssl/lib/hpux64/libssl.a * • /opt/openssl/0.9.8/lib/hpux64/libssl.a • /opt/openssl/0.9.7/lib/hpux64/libssl.0.9.8g.a
	/opt/openssl/0.9.8/lib/hpux64/libcrypto.0.9.8g.a	<ul style="list-style-type: none"> • /usr/lib/hpux64/libcrypto.a * • /opt/openssl/lib/hpux64/libcrypto.a * • /opt/openssl/0.9.8/lib/hpux64/libcrypto.a • /opt/openssl/0.9.7/lib/hpux64/libcrypto.0.9.8g.a

Table 1-5 OpenSSL A.00.09.08d Intel Itanium Libraries *(continued)*

Library	Library Name/Location	Symbolic Link
64-bit shared	/opt/openssl/0.9.8/lib/hpux64/libssl.so.1	<ul style="list-style-type: none"> • /usr/lib/hpux64/libssl.so * • /usr/lib/hpux64/libssl.so.1 • /opt/openssl/lib/hpux64/libssl.so * • /opt/openssl/lib/hpux64/libssl.so.1 • /opt/openssl/0.9.8/lib/hpux64/libssl.so • /opt/openssl/0.9.7/lib/hpux64/libssl.so.1
	/opt/openssl/0.9.8/lib/hpux64/libcrypto.so.1	<ul style="list-style-type: none"> • /usr/lib/hpux64/libcrypto.so * • /usr/lib/hpux64/libcrypto.so.1 • /opt/openssl/lib/hpux64/libcrypto.so * • /opt/openssl/lib/hpux64/libcrypto.so.1 • /opt/openssl/0.9.8/lib/hpux64/libcrypto.so • /opt/openssl/0.9.7/lib/hpux64/libcrypto.so.1



NOTE: Symbolic links marked * are applicable only if the default version is OpenSSL A.00.09.08g.

Client and server programs can use these OpenSSL C library functions to add SSL protocol support, and to create and accept SSL network

The openssl Command-Line Tool

The `openssl` command-line tool is an interactive tool that enables you to execute cryptographic functions. It supports the following features:

- Creating and viewing secret keys
- Encrypting or decrypting files using secret-key ciphers
- Calculating message digests for files
- Creating and viewing RSA, DSA, and DH public keys
- Encrypting or decrypting a file using a public key or private key, respectively
- Creating X.509 certificates, certificate requests, and Certificate Revocation Lists (CRL)
- Managing the Certificate Authority (CA)

Strong Random Number Generator for HP-UX 11i v1

OpenSSL A.00.09.07m requires a strong random number generator to provide secure and non reproducible keys and certificates.

OpenSSL A.00.09.07m looks for the random number generator in the following order:

1. /dev/urandom
2. /dev/random
3. /opt/openssl/prngd/prngd

If none of these random number generators are available on the system, OpenSSL returns an error while executing cryptographic functions. To prevent this situation, OpenSSL for HP-UX 11i v1 includes the /opt/openssl/prngd/prngd random number generator. The prngd server reads HP-UX commands from the prngd.conf file, computes random numbers based on certain parameters, and writes the computed random numbers to an HP-UX socket located in the /var/run/egd-pool directory. OpenSSL functions can connect to and read random numbers from this socket. The HP-UX 11i v2 and HP-UX 11i v3 operating systems contain /dev/random by default; therefore, it does not require /opt/openssl/prngd/prngd. Random number generation using /dev/urandom or /dev/random is faster than using /opt/openssl/prngd/prngd. HP-UX 11i v1 users can download /dev/random from the following location: <http://www.software.hp.com>

Automatically Generated Self-Signed Host Certificate

An SSL-enabled server must be identified by a host certificate. A certificate also identifies the network host, the name and ID of the Certificate Authority (CA), and expiry date of the certificate. Before you can deploy an SSL-enabled server for production, it must acquire a certificate signed by a legitimate CA. However, for testing purposes the certificate can be self-signed, that is, signed by the application generating the certificate. Setting up a certificate hierarchy can be time-consuming. If a self-signed certificate is available, you can direct your SSL server to this certificate during testing. OpenSSL automatically generates a self-signed host certificate and private key. The host certificate is stored as /opt/openssl/certs/host.pem and the private key of the host certificate is saved as /opt/openssl/private/hostkey.pem. The subject name of the certificate is as follows:

```
C=US, ST=CA, L=City, O=Company,  
CN=localhost/emailAddress=www@localhost
```

You can also generate a self-signed host certificate using the following command:

```
$ openssl req -new -x509 -out /opt/openssl/certs/host.pem  
-keyout /opt/openssl/private/hostkey.pem -nodes  
-subj /C=US/ST=CA/L=City/O=Company/CN=localhost/emailAddress=www@localhost
```

Defects Fixed in OpenSSL Versions A.00.09.07m and A.00.09.08g

Several cleanup-related changes, more stringent error checking, and general fixes in OpenSSL Open Source versions 0.9.7m and 0.9.8g are included in OpenSSL versions A.00.09.07m and A.00.09.08g. A complete list of defect fixes is available in the OpenSSL Changelog at:

<http://www.openssl.org/news/changelog.html>

Additional Information on OpenSSL Defects and Fixes

Additional information on OpenSSL defects and fixes is available in the following locations:

- For general information on security vulnerabilities for different products, refer to the following Website:

<http://www.secunia.org>

Search for the keyword “OpenSSL” to locate security vulnerabilities for OpenSSL.

- To obtain source code level information about specific patches, refer to the following Website:

<http://cvs.openssl.org>

Known Problems

Following are the known problems in this version of OpenSSL:

In OpenSSL A.00.09.07m and OpenSSL A.00.09.08g After `ERR_free_strings()` is called, the `ERR_load_crypto_strings()` function no longer works. It does not load error strings, and `ERR_error_string()` does not return an error message.

In OpenSSL A.00.09.07m

- EVP with AES-cfb1, DES-cfb1 and Camellia-cfb1 does not work properly.

In OpenSSL A.00.09.08g

- Datagram-TLS protocol support for securing datagram transport (UDP for instance) fails and `s_server` dumps core. HP is currently working to resolve this issue.

Compatibility Information and Installation Requirements

This section lists the system and patch requirements for OpenSSL A.00.09.07m and A.00.09.08g.

System Requirements

Table 1-6 specifies the minimum system requirements for installing OpenSSL A.00.09.07m and A.00.09.08g.

Table 1-6 System Requirements for Installing OpenSSL A.00.09.07m and A.00.09.08g

Component	Requirement
Operating system	<ul style="list-style-type: none"> HP-UX 11i v1 HP-UX 11i v2 HP-UX 11i v3
Hardware requirement	<ul style="list-style-type: none"> HP 9000 systems HP Integrity systems
Disk space requirement	100 MB
Software availability in native languages	English only

Patch Requirements

HP has tested the OpenSSL A.00.09.07m and A.00.09.08g software in test environments with the Support Plus media listed in Table 1-7.

HP-UX 11i v1 customers must install the GOLDQPK11i QPK that is delivered on the Support Plus media. No patches are required for installing OpenSSL on HP-UX 11i v2 and HP-UX 11i v3.

Table 1-7 Patch Bundle Required for HP-UX 11i v1 Customers

Operating System	Required Support Plus Media		
	Date	Release	Part #
HP-UX 11i v1	March 2002	SP-0203	5012-0049

For more information about the Support Plus media, refer to the IT Resource Center at: <http://www.itrc.hp.com>



NOTE: OpenSSL has been tested with the above QPK bundle. You can also choose to use the latest QPK bundle as these bundles are cumulative in nature.

HP recommends that you use OpenSSL A.00.09.07m and A.00.09.08g with the `libc` patch listed in Table 1-8.

Table 1-8 Recommended libc Patch Bundles

Operating System	Recommended libc Patch Bundle
HP-UX 11i v1	PHCO_28427

Installing OpenSSL

To install OpenSSL, complete the following steps:

1. Log in as superuser.
2. Insert the software CD into the appropriate drive if you are installing from the Application Release CD. If you are downloading the software package from the Software Depot, download the depot and follow the instructions provided in the installation page for OpenSSL.
3. To install openssl and fips_1_1_2 products, enter the following command:

```
# swinstall -s /tmp/OpenSSL*.depot OpenSSL
```

To install openssl product only, enter the following command:

```
# swinstall -s /tmp/OpenSSL*.depot openssl
```

To install the FIPS Capable OpenSSL product (FIPS 1.1.2 object module and 0.9.7m based FIPS Compatible OpenSSL) only, enter the following command:

```
# swinstall -s /tmp/OpenSSL*.depot fips_1_1_2
```

The swinstall command installs the OpenSSL software in the /opt/openssl directory. It places the sample codes in the /opt/openssl/src and /opt/openssl/fips/0.9.7/src directories.



IMPORTANT: You cannot install OpenSSL A.00.09.07m or A.00.09.08g on a system containing HP-UX Internet Express OpenSSL 0.9.7c. If HP-UX Internet Express OpenSSL 0.9.7c software is installed on your system, you must remove it before installing OpenSSL A.00.09.07m or A.00.09.08g.



NOTE: On HP-UX 11i v1 systems without /dev/random, the prngd server is started automatically. A random number generator (either /dev/random or prngd) is required to generate keys and certificates.

Using the openssl Command-Line Tool

This section lists the options supported by the openssl command-line tool and discusses procedures to create an RSA key, a certificate request, and a self-signed certificate.

Options

Table 1-9 describes the openssl command-line tool options.

Table 1-9 The openssl Command-Line Options

Option Name	Description
ca	CA management
crl	CRL management
dgst	Message digest calculation
dsa	DSA data management
enc	Encoding with ciphers
gensa	Generation of DSA parameters
genrsa	Generation of RSA parameters
req	X.509 Certificate Signing Request (CSR) management
rsa	RSA data management
verify	X.509 certificate verification
x509	X.509 certificate data management

For more information on `openssl` command-line options, refer to *openssl(1)*.

Using openssl

This section explains the use of the `openssl` command-line tool with examples. For more information, refer to the *openssl(1)* manpage.

Creating an RSA Key

Following is the syntax to create an RSA public and private key pair:

```
$ openssl genrsa -out <filename> <bits>
```

where:

<bits> Specifies the size of the key.

<filename> Specifies the file name where the key must be stored.

To create an RSA public and private key pair, use the following command:

```
$ openssl genrsa -out key.pem 1024
```

This command creates a 1024-bit key pair and stores it in a file called `key.pem`. The `<bits>` parameter is optional. The default key size is 512 bits.

Creating a Password-Protected RSA Key Pair

Following is the syntax to create a password-protected private RSA key pair:

```
$ openssl genrsa -<encryption-algorithm> -out <filename> <bits>
```

where:

`<encryption-algorithm>` Specifies the algorithm to be used for encrypting the private key (using a password supplied by the user).

To create a password-protected private RSA key pair, use the following command:

```
$ openssl genrsa -des3 -out key.pem 1024
```

When you execute this command, the program prompts for a pass phrase (password), which is used to encrypt the key file (with the specified algorithm). You must enter the correct pass phrase to view the key.

Viewing an RSA Key Pair

Following is the syntax to view an RSA key pair:

```
$ openssl rsa -in <filename> -noout -text
```

To view the modulus, exponent, and prime key values use the following command:

```
$ openssl rsa -in key.pem -noout -text
```

This command displays the key pair stored in the `key.pem` file. If the key pair stored in `key.pem` is encrypted, then this command prompts the user for the pass phrase.

Creating an RSA Certificate Request

Following is the syntax to create a new certificate request:

```
$ openssl req -new -nodes -out <filename> -keyout <keyfile>
-subj <subject>
```

where:

`<filename>` Specifies the file to which the certificate request is written.

`<keyfile>` Specifies the file to which the RSA public and private key pair for the certificate is written.

`<subject>` Specifies the subject name of the certificate.

To create an RSA certificate request, use the following command:

```
$ openssl req -new -nodes -out cert.txt -keyout key.pem
-subj "/C=US/ST=CA/L=CITY/CN=localhost/emailAddress=root@localhost"
```

Creating a Self-Signed Certificate

Following is the syntax to create a self-signed certificate:

```
$ openssl req -new -nodes -x509 -out <filename> -keyout <keyfile>
-days <numdays> -subj <subject>
```

where:

`-x509` Indicates a self-signed certificate.

`<numdays>` Indicates the number of days for which the certificate is valid.

To create a self-signed certificate, use the following command:

```
$ openssl req -new -nodes -x509 -out cert.pem -keyout key.pem
-days 365 -subj "/C=US/ST=CA/L=City/CN=localhost/emailAddress=root@localhost"
```

OpenSSL Resources

This section provides a list of sources from which you can obtain the OpenSSL software, and pointers to obtain information about OpenSSL technology.

Getting the OpenSSL Software

You can obtain OpenSSL A.00.09.07m and A.00.09.08g software from the following sources:

- HP Software Depot at: <http://www.software.hp.com>
- HP-UX Operating Environments (OEs)



TIP: The most recent version of OpenSSL is available at: <http://www.software.hp.com>

Learning About OpenSSL Technology

A large volume of information exists on the Internet about OpenSSL technology. HP recommends that you learn more about OpenSSL by reading O'Reilly's book *Network Security with OpenSSL: Cryptography for Secure Communications* by John Viega, Matt Messier, and Pravir Chandra. You can order this book from <http://www.oreilly.com/>

You can also learn about the OpenSSL technology at the following links:

- OpenSSL Website at: <http://www.openssl.org/>
- OpenSSL FAQ at: <http://www.openssl.org/support/faq.html>
- OpenSSL mailing list at: <http://marc.theaimsgroup.com/?l=openssl-users>
- The Transport Layer Security (TLS) Internet Engineering Task Force (IETF) Working Groups at: <http://www.ietf.org/html.charters/wg-dir.html#Security%20Area>
- OpenSSL APIs at: <http://www.opensslbook.com/api/index.html>

OpenSSL A.00.09.07m.008, A.00.09.07m.009, and A.00.09.08g.007 Release Notes is available at the following locations:

- The HTML and pdf versions are available at: <http://docs.hp.com/hpux/internet/index.html> (Internet and Security Solutions).
- A text version of the README.hp readme file in the /opt/openssl directory.

Frequently Asked Questions (FAQs)

Following are questions frequently asked about OpenSSL.

- 1 What does OpenSSL do? Why do I need it?
OpenSSL offers an advanced level of security using the SSL/TLS protocols. Client-server applications that send and receive data over a network are open to a range of vulnerabilities. They can use SSL/TLS to implement privacy (through encryption), tamper-proofing (through message digests) and non-repudiation (through certificates and digital signatures).
- 2 What is the openssl command-line tool? Why do I need it?

The OpenSSL libraries (`libssl` and `libcrypto` - the 32 and 64-bit versions of the static and shared libraries) define the OpenSSL product. The `openssl` command-line tool is an easy way for you to quickly execute functions (for example, create certificates) without having to write a new application for that purpose.



NOTE: The `openssl` command-line tool is a 32-bit application. It uses the 32-bit static OpenSSL libraries.

- 3 There are several flavours of libraries available in OpenSSL A.00.09.07m and OpenSSL A.00.09.08g. What are they? How do I know when to use which library? Use the OpenSSL A.00.09.07m and OpenSSL A.00.09.08g libraries for 32-bit and 64-bit applications. Both the 32-bit and 64-bit versions of the libraries are provided. For a list of all the library files, see “OpenSSL Libraries” (page 8). You can also choose to create user applications using either a static library or a shared library. In addition, OpenSSL A.00.09.08g contains libraries that support hardware ENGINES.
- 4 How do I switch between OpenSSL A.00.09.07m and OpenSSL A.00.09.08g? During installation, the depot installs OpenSSL A.00.09.07m and OpenSSL A.00.09.08g in the `/opt/openssl/0.9.7` and `/opt/openssl/0.9.8` directories, respectively. These directories contain binaries, libraries, manpages, and other files specific to each version of OpenSSL. The `/opt/openssl/switchversion.sh` script switches between these two versions. To change the version of OpenSSL, execute the script as follows:

```
$ /opt/openssl/switchversion.sh
```

You can also choose to switch the `openssl.cnf` file based on the version of OpenSSL. However this is not necessary.
- 5 What is the performance as compared to the Open Source version 0.9.7m or 0.9.8g respectively? The two products have the same base source code. There is no difference in performance, other conditions remaining the same. However, the performance of several `openssl` library functions is dictated by the random number generator on the system. The `/dev/urandom` and `/dev/random` devices perform better than `prngd`. You can download `/dev/random` at:
<http://software.hp.com>
- 6 Does installing OpenSSL require a kernel rebuild? No. OpenSSL contains application libraries and a command-line tool. It does not require a kernel rebuild or system reboot.
- 7 How can I install OpenSSL A.00.09.07m or A.00.09.08g? You can install OpenSSL A.00.09.07m or A.00.09.08g from the application CD or the Web using the `swinstall` command.
- 8 How can I uninstall OpenSSL A.00.09.07m or A.00.09.08g?

Use the following command to uninstall OpenSSL:

```
$ swremove OpenSSL
```

- 9 I have already got the supported version A.00.09.07l on my HP-UX system, and I am quite happy with it. Why do I need to move to A.00.09.07m ?

This new version of OpenSSL contains several bug fixes, but most importantly, it has a few critical fixes that have been well publicized at the OpenSSL site. HP recommends that you upgrade to OpenSSL A.00.09.07m even if you are not affected by these defects.

- 10 If I do move my existing applications to this new version, will I need to rebuild the application? Do you guarantee binary compatibility of my applications with this new version?

If you have built your application with either OpenSSL version A.00.09.07d, A.00.09.07e, A.00.09.07i, or A.00.09.07l for a given operating system (HP-UX 11i v1, 11i v2, or 11i v3), you do not have to rebuild your application. HP does not guarantee binary compatibility of applications built with any other version of OpenSSL.

- 11 I installed the HP-UX OpenSSL 0.9.7c version late in 2003, and it is still on my system. What is the history (and support details) for that version of OpenSSL, and how does that compare with OpenSSL A.00.09.07m and A.00.09.08g?

OpenSSL A.00.09.07m and A.00.09.08g contain a precompiled version of the OpenSSL Open Source versions 0.9.7m and A.09.08g. Additionally, OpenSSL A.00.09.07m and A.00.09.08g contain the prngd random number generator and a self-signed host certificate. OpenSSL A.00.09.07m and A.00.09.08g is built to install and uninstall using the SD-UX utility, and is only for customers using the HP-UX operating system. HP offers full support for OpenSSL A.00.09.07m and A.00.09.08g.

The Internet Express OpenSSL 0.9.7c product is an older version, and contains a precompiled version of the OpenSSL Open Source version 0.9.7c. It is also built to install and uninstall using the SD-UX utility, and is meant for customers using the HP-UX operating system. However, HP does not offer support for Internet Express OpenSSL 0.9.7c. This version is the last Internet Express version of OpenSSL released

- 12 How many fully supported releases of OpenSSL have been done on HP-UX so far?

Apart from this version of OpenSSL, HP fully supports the following versions of OpenSSL:

- OpenSSL A.00.09.07d (based on OpenSSL Open Source version 0.9.7d)
- OpenSSL A.00.09.07e (based on OpenSSL Open Source version 0.9.7e)
- OpenSSL A.00.09.07i (based on OpenSSL Open Source version 0.9.7i)
- OpenSSL A.00.09.07l (based on OpenSSL Open Source version 0.9.7l)

A few patch updates are available for these releases of OpenSSL. A patch release typically has the same base version number (for example, A.00.09.07m) with a patch number appended to it (for example, A.00.09.07m.009).

- 13 I have HP-UX Internet Express OpenSSL 0.9.7c installed on my system. Will installing OpenSSL A.00.09.07m and A.00.09.08g automatically uninstall HP-UX Internet Express OpenSSL 0.9.7c?

No. The HP-UX Internet Express OpenSSL 0.9.7c and OpenSSL A.00.09.07m and A.00.09.08g product depots have a conflict with the product and bundle names. If you have the HP-UX Internet Express OpenSSL product installed and want to upgrade to OpenSSL A.00.09.07m or A.00.09.08g, you must first uninstall the HP-UX Internet Express OpenSSL product. If you attempt to install OpenSSL A.00.09.07m or A.00.09.08g on a system without removing the HP-UX Internet Express OpenSSL product, the OpenSSL A.00.09.07m and A.00.09.08g installation fails with an error message. If you have HP-UX Internet Express OpenSSL 0.9.7c installed on your system, use the following command to remove it:

```
# swremove ixOpenSSL
```

- 14 I have already built Open Source OpenSSL 0.9.7m or A.0.9.8g by downloading the source code directly from <http://www.openssl.org>. Now I want to upgrade to OpenSSL A.00.09.07m or A.00.09.08g. What must I do? Do I have to remove the preexisting OpenSSL product from my system?

You may have a conflict depending on the location of OpenSSL 0.9.7m and 0.9.8g on your system. HP recommends that you uninstall OpenSSL 0.9.7m or 0.9.8g before installing OpenSSL A.00.09.07m and A.00.09.08g.

- 15 Will HP support recompiled versions of OpenSSL A.00.09.07m or OpenSSL A.00.09.08g?

HP does not support recompiled versions of OpenSSL A.00.09.07m and A.00.09.08g. The source code is provided only for reference.

- 16 Why are idea, rc5, and mdc2 algorithms not configured in OpenSSL A.00.09.07m and A.00.09.08g?

The idea, rc5, and mdc2 crypto algorithms have patent issues. HP does not redistribute any software using these algorithms.

- 17 How do I find out if I am running OpenSSL A.00.09.07m or OpenSSL A.00.09.08g?

Use the `what` command to find out whether you are running OpenSSL A.00.09.07m or OpenSSL A.00.09.08g.

Example 1-1 If you are running OpenSSL A.00.09.07m.020 on HP-UX 11i v1

```
# what /usr/bin/openssl
/usr/bin/openssl:
  $OpenSSL A.00.09.07m.020 $
  $OpenSSL A.00.09.07m.020 $
  $OpenSSL A.00.09.07m.020 $
```

Example 1-2 If you are running OpenSSL A.00.09.08g.019 on HP-UX 11i v3

```
# what /usr/bin/openssl
/usr/bin/openssl:
  $OpenSSL A.00.09.08g.019 $
  $OpenSSL A.00.09.08g.019 $
  $OpenSSL A.00.09.08g.019 $
```

- 18 How do I find out whether I am running Internet Express OpenSSL, OpenSSL A.00.09.07m, OpenSSL A.00.09.08g or the Open Source version of OpenSSL? Run the `what` command to display the OpenSSL bundle installed on your system. Following are some examples:

Example 1-3 When an old version of OpenSSL from Internet Express is installed on the system

```
# what /usr/bin/openssl
OpenSSL A.02.00-0.9.7c
```

Example 1-4 When OpenSSL A.00.09.07m.021 is installed on an HP-UX 11i v2 operating system

```
# what /usr/bin/openssl
/usr/bin/openssl:
  $OpenSSL A.00.09.07m.021 $
  $OpenSSL A.00.09.07m.021 $
  $OpenSSL A.00.09.07m.021 $
```

Example 1-5 You do not have an HP-UX depot installed but have downloaded the source code and built the product yourself

```
# what /usr/bin/openssl
```

The output of the `what` command depends on what you defined for the `$what` string in the source code before you built the product.

- 19 A lot of information is available for OpenSSL-enabler products, such as Stunnel. How is Stunnel different from my OpenSSL A.00.09.07m or A.00.09.08g installation?

Consider a client-server application sending and receiving unencrypted data over the network. The following methods describe how to use OpenSSL technology to encrypt client/server communication:

- To implement secure communication, you can modify client and server code using OpenSSL functions. This method gives you the highest degree of flexibility and control regarding the features and how you implement them.
- You can set up an Stunnel-based environment in which data from the client application is sent to an Stunnel client instead of to the server. The Stunnel client encrypts the data using OpenSSL technology and sends encrypted data to the Stunnel server. The Stunnel server decrypts the data and sends the original data to the target server application. The same process is followed when data is sent from the server to the client application. This approach enables you to secure your client-server application without changing the source code, but limits you to the features offered by the Stunnel environment.

These are the two distinct choices available to a user application environment that wants to SSL-encrypt its client-server communication. Both choices are valid. Direct use of the OpenSSL library clearly provides more options.

20 What is the FIPS relationship to the OpenSSL API?

The FIPS object module is designed for use with the OpenSSL API. Applications linked with the FIPS object module and with the separate OpenSSL libraries can use both the FIPS-validated cryptographic functions of the FIPS object module and the high level functions of OpenSSL.

The FIPS object module is the special monolithic object module built from the special source distribution identified in the Security Policy. It is not the same as the OpenSSL product or any specific official OpenSSL distribution release.

A version of the OpenSSL product that is suitable for reference by an application along with the FIPS object module is a FIPS compatible OpenSSL which links against FIPS Object Module 1.1.2.

When the FIPS object module and a FIPS compatible OpenSSL are separately built and installed on a system, the combination is referred to as a FIPS capable OpenSSL.

21 What kind of cryptographic algorithms can be used in FIPS mode?

Table 1-10 lists the cryptographic algorithms that can and cannot be used in FIPS mode.

Table 1-10 Cryptographic Algorithms that Can be Used in FIPS mode and Standard OpenSSL Mode

Algorithm Type	Algorithm	Standard OpenSSL	FIPS	Usage
Asymmetric keys	RSA	Supported	Supported	<ul style="list-style-type: none"> • Key agreement • Digital signature • Encryption/Decryption
	DSA			Digital signature
	DH			Key agreement
Symmetric keys	AES	Supported	Supported	Encryption/Decryption
	Blowfish		Not supported	
	CAST		Not supported	
	DES		Not supported	
	DES3		Supported	
	DESX		Not supported	
	RC2		Not supported	
	RC4		Not supported	
HMAC	HMAC-MD2	Supported	Not supported	<ul style="list-style-type: none"> • Module integrity • Code integrity • Message integrity
	HMAC-MD4			
	HMAC-MD5			
	HMAC-RMD160			
	HMAC-SHA			
	HMAC-SHA1		Supported	
Hashing	MD2	Supported	Not supported	Hashing
	MD4			
	MD5			
	RMD160			
	SHA			
	SHA1		Supported	