Hewlett Packard
Enterprise

OpenSSL A.01.00.02h.001 Release Notes

HP-UX 11i v3

Part Number: 828902-005
Published: June 2016
Edition: 1

Acknowledgements

Contents

HPE secure development lifecycle

Starting with HP-UX 11i v3 March 2013 update release, HPE secure development lifecycle provides the ability to authenticate HP-UX software. Software delivered through this release has been digitally signed using HPE's private key. You can now verify the authenticity of the software before installing the products, delivered through this release.

To verify the software signatures in signed depot, the following products must be installed on your system:

- `B.11.31.1303` or later version of SD (Software Distributor)
- `A.01.02.00` or later version of HP-UX Whitelisting `(WhiteListInf)`

To verify the signatures, run: `/usr/sbin/swsign -v -s <depot_path>`.

For more information, see *Software Distributor documentation* at **http://www.hpe.com/info/sd-docs**.

NOTE: Ignite-UX software delivered with HP-UX 11i v3 March 2014 release or later supports verification of the software signatures in signed depot or media, during cold installation.

For more information, see *Ignite-UX documentation* at **http://www.hpe.com/info/ignite-ux-docs**.

1 OpenSSL A.01.00.02h.001

This document contains the most recent product information for OpenSSL A.01.00.02h.001 supported on HP-UX 11i v3. This document contains the following information:

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

1.1 Announcement

This version of OpenSSL is based on the open source OpenSSL A.01.00.02h, A.00.09.08zf, and A.00.09.07m products. This bundle contains the following:

- OpenSSL A.01.00.02h in the `/opt/openssl/1.0` directory.
  - For both IA64 and PA machine architecture.
- OpenSSL A.00.09.08zf in the `/opt/openssl/0.9.8` directory
  - For both IA64 and PA machine architecture.
- OpenSSL A.00.09.07m in the `/opt/openssl/0.9.7` directory
  - For both IA64 and PA machine architecture.
- FIPS capable OpenSSL (based on open source OpenSSL version 1.0.2h and linked against with FIPS 2.0.5 module) in the `/opt/openssl/fips/1.0` directory.
  - For IA64 machine architecture only.
- FIPS Capable OpenSSL (based on open source OpenSSL version 0.9.8zf and linked against with FIPS 1.2 module) in the `/opt/ openssl/fips/0.9.8` directory.
- FIPS Capable OpenSSL (based on opensource OpenSSL version 0.9.7m and linked against with FIPS 1.1.2 module) in the `/opt/ openssl/fips/0.9.7` directory.

The default version of OpenSSL that is enabled on IA64 and PA hardware is OpenSSL A.01.00.02h. Use the `/opt/openssl/switchversion.sh` script to switch the default version of OpenSSL to other versions. 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.01.00.02h, A.00.09.08zf, and A.00.09.07m offer a general-purpose cryptography library and implementation of the Secure Sockets Layer and Transport Layer Security protocols.

This is the release of HP-UX OpenSSL A.01.00.02h. The full version string of the product is:

A.01.00.02h.001.

NOTE:

- OpenSSL version 1.0.2 is not backward compatible with OpenSSL A.00.09.08zf or earlier versions. For more information, see the FAQ section around “binary compatibility” to check if you are affected.

OpenSSL is available as web upgrades. The software bits are available at: **https://www.hpe.com/support/softwaredepot**.

OpenSSL on IA64 hardware has been build with the following hardware options:

```
./config threads zlib shared no-rc5 no-idea no-krb5 no-mdc2
  --openssldir=/opt/openssl --with-zlib-include=<zlib_include_dir>
  -Wl,+nodefaultrpath
```

OpenSSL A.01.00.02h on PA hardware has been built with the following options:
For 32 bit build

```
./Configure threads zlib shared no-rc5 no-idea no-krb5 no-mdc2
--openssldir=<OpenSSL installation dir>
--with-zlib-include=<location of zlib directory>
-D_REENTRANT hpux-parisc2-cc -Wl,+nodefaultrpath
```

For 64 bit build

```
./Configure threads zlib shared no-rc5 no-idea no-krb5 no-mdc2
--openssldir=<OpenSSL installation dir>
--with-zlib-include=<location of zlib directory>
-D_REENTRANT hpux64-parisc2-cc -Wl,+nodefaultrpath
```

OpenSSL A.00.09.08zf has been built with the following options:

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

OpenSSL A.00.09.07m has been built with the following options:

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

FIPS Capable OpenSSL (based on OpenSSL A.01.00.02h and linked against FIPS-2.0.5 module) is built with the following options:

```
IA:
32 bit build:
./config fips threads zlib shared no-rc5 no-idea no-krb5 no-mdc2
  --openssldir=/opt/openssl --with-zlib-include=<zlib_include_dir>
  --with-fipslibdir="/usr/local/ssl/lib"
  -Wl,+nodefaultrpath -Wl,+b/opt/openssl/fips/1.0/lib/hpux32 -Wl,+rpathfirst
64 bit build:
./config fips threads zlib shared no-rc5 no-idea no-krb5 no-mdc2
  --openssldir=/opt/openssl --with-zlib-include=<zlib_include_dir>
  --with-fipslibdir="/usr/local/ssl/lib"
  -Wl,+nodefaultrpath -Wl,+b/opt/openssl/fips/1.0/lib/hpux64 -Wl,+rpathfirst
```

FIPS Capable OpenSSL (based on OpenSSL A.00.09.08zf and linked against FIPS-1.2 module) is built with the following options:

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

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.
`-Wl, <options>`	Linkers options to be passed while building OpenSSL.

For more information about how to build FIPS Capable OpenSSL A.01.00.02h on HPUX, see README.hp in `/opt/openssl/fips/1.0/src` directory.

For more information about how to build Non FIPS Capable OpenSSL A.01.00.02h on HPUX, see the README.hp in `/opt/openssl/1.0/src` directory.

For more information about how to build OpenSSL A.00.09.07m and A.00.09.08zf on HPUX, see the README.hp embedded within `/opt/openssl/0.9.7/src/openssl-0.9.7m.tar.gz` and `/opt/openssl/0.9.8/src/openssl-0.9.8zf.tar.gz`.

1.2 OpenSSL features

OpenSSL supports the following security features:

- Ciphers
- Digests
- Public key
- Certificates
- Encoding
- Federal Information Processing Standard (FIPS)

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

1.2.1 Ciphers

A cipher algorithm is a mechanism used to encrypt or decrypt a message. OpenSSL supports the following ciphers:

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

△ **CAUTION:** DES has been cracked (data encoded by DES has been decoded by a third party). HPE 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)
- Camellia (A.01.00.01j and above)
- Seed (A.01.00.01j and above)

1.2.2 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 supports the following message digest algorithms:

- Hashed Message Authentication Code (HMAC)
- Message Digest 2 (MD2) algorithm (not available in Version A.01.00.01j or later)
- 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)
- Secure Hash Algorithm 2 (SHA2)

  NOTE: OpenSSL 0.9.7 version does not support SHA2.

- Whirlpool (Version A.01.00.01j and above)

1.2.3 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 supports the following public key encryption methods:

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

1.2.4 Certificates

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

OpenSSL supports the following digital certificates:

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

1.2.5 Encoding

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

- ASN.1—Abstract Syntax Notation One.
- Distinguished Encoding Rules (DER)—Stores 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.

1.2.6 FIPS

FIPS capable open source OpenSSL version A.01.00.02h based on “FIPS Object Module” version 2.0.5 is provided on HP-UX Integrity systems. For more information and usage of FIPS capable OpenSSL, see `/opt/openssl/fips/1.0/README.hp`.

IMPORTANT: The FIPS code is certified only if it is identical with the source code released on the OpenSSL website. In the event of a security vulnerability, HPE 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, HPE will wait until the openssl.org releases a new FIPS 140-2 certified FIPS module before updating the HP-UX OpenSSL product with the new FIPS code.

1.3 What is new in OpenSSL A.01.00.02h

There are several enhancements made between different versions of OpenSSL. A complete list of enhancements and fixes in each version is available in the OpenSSL Changelog at **http://www.openssl.org/news/changelog.html**.

The OpenSSL release notes (provided by **www.openssl.org** at **https://www.openssl.org/news/openssl-1.0.2-notes.html**) lists the major enhancements and fixes made between different versions of OpenSSL.

1.4 OpenSSL components

OpenSSL contains the following components:

- OpenSSL libraries.
- The `openssl` command-line tool.
- Automatically generated self-signed host certificate.

1.4.1 OpenSSL libraries

OpenSSL contains 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 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 is installed on the system. These symbolic links are listed in the following tables:

Table 1 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`	• `/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`	• `/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`

Table 1 OpenSSL A.00.09.07m PA-RISC libraries *(continued)*

Library	Library Name/Location	Symbolic Link
32-bit shared	/opt/openssl/0.9.7/lib/libssl.sl.0	• /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	• /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	• /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	• /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
64-bit shared	/opt/openssl/0.9.7/lib/pa20_64/libssl.sl.0	• /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	• /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 2 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	• /usr/lib/hpux32/libssl.a * • /opt/openssl/lib/hpux32/libssl.a * • /opt/openssl/0.9.7/lib/hpux32/libssl.a • /opt/openssl/1.0/lib/hpux32/libssl.0.9.7m.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	• /usr/lib/hpux32/libcrypto.a * • /opt/openssl/lib/hpux32/libcrypto.a * • /opt/openssl/0.9.7/lib/hpux32/libcrypto.a • /opt/openssl/1.0/lib/hpux32/libcrypto.0.9.7m.a • /opt/openssl/0.9.8/lib/hpux32/libcrypto.0.9.7m.a
32-bit shared	/opt/openssl/0.9.7/hpux32/libssl.so.0	• /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/1.0/lib/hpux32/libssl.so.0
	/opt/openssl/0.9.7/hpux32/libcrypto.so.0	• /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 • /opt/openssl/1.0/lib/hpux32/libcrypto.so.0
64-bit static	/opt/openssl/0.9.7/lib/hpux64/libssl.0.9.7m.a	• /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/1.0/lib/hpux64/libssl.0.9.7m.a
	/opt/openssl/0.9.7/lib/hpux64/libcrypto.0.9.7m.a	• /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 • /opt/openssl/1.0/lib/hpux64/libcrypto.0.9.7m.a

Table 2 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	• /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/1.0/lib/hpux64/libssl.so.0
	/opt/openssl/0.9.7/lib/hpux64/libcrypto.so.0	• /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 • /opt/openssl/1.0/lib/hpux64/libcrypto.so.0

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

Table 3 OpenSSL A.00.09.08zf PA-RISC libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/0.9.8/lib/libssl.0.9.8zf.a	• /usr/lib/libssl.a * • /opt/openssl/lib/libssl.a * • /opt/openssl/0.9.8/lib/libssl.a • /opt/openssl/0.9.7/lib/libssl.0.9.8zf.a
	/opt/openssl/0.9.8/lib/libcrypto.0.9.8zf.a	• /usr/lib/libcrypto.a * • /opt/openssl/lib/libcrypto.a * • /opt/openssl/0.9.8/lib/libcrypto.a • /opt/openssl/0.9.7/lib/libcrypto.0.9.8zf.a

Table 3 OpenSSL A.00.09.08zf PA-RISC libraries *(continued)*

Library	Library Name/Location	Symbolic Link
32-bit shared	/opt/openssl/0.9.8/lib/libssl.sl.1	• /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	• /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.0.9.8zf.a	• /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.0.9.8zf.a
	/opt/openssl/0.9.8/lib/pa20_64/libcrypto.0.9.8zf.a	• /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.0.9.8zf.a
64-bit shared	/opt/openssl/0.9.8/lib/pa20_64/libssl.sl.1	• /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	• /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/openssl/0.9.7/lib/pa20_64/libcrypto.sl.1

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

Table 4 OpenSSL A.00.09.08zf Intel Itanium® libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/0.9.8/lib/ hpux32/ libssl.0.9.8zf.a	• /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.0.9.8zf.a • /opt/openssl/1.0/lib/hpux32/ libssl.0.9.8zf.a
	/opt/openssl/0.9.8/lib/ hpux32/ libcrypto.0.9.8zf.a	• /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.0.9.8zf.a • /opt/openssl/1.0/lib/hpux32/ libcrypto.0.9.8zf.a
32-bit shared	/opt/openssl/0.9.8/ hpux32/ libssl.so.1	• /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/1.0/lib/hpux32/libssl.so.1
	/opt/openssl/0.9.8/ hpux32/ libcrypto.so.1	• /usr/lib/hpux32/libcrypto.so * • /usr/lib/hpux32/libcrypto.so.1 • /opt/openssl/lib/hpux32/libcrypto.so * • /opt/openssl/ lib/hpux32/libcrypto.so.1 • /opt/openssl/0.9.8/lib/hpux32/libcrypto.so • /opt/openssl/0.9.7/lib/hpux32/ libcrypto.so.1 • /opt/openssl/1.0/lib/hpux32/libcrypto.so.1

Table 4 OpenSSL A.00.09.08zf Intel Itanium® libraries *(continued)*

Library	Library Name/Location	Symbolic Link
64-bit static	/opt/openssl/0.9.8/lib/ hpux64/ libssl.0.9.8zf.a	• /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.0.9.8zf.a • /opt/openssl/1.0/lib/hpux64/ libssl.0.9.8zf.a
	/opt/openssl/0.9.8/lib/ hpux64/ libcrypto.0.9.8zf.a	• /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.0.9.8zf.a • /opt/openssl/1.0/lib/hpux64/ libcrypto.0.9.8zf.a
64-bit shared	/opt/openssl/0.9.8/lib/ hpux64/ libssl.so.1	• /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/1.0/lib/hpux64/libssl.so.1
	/opt/openssl/0.9.8/lib/ hpux64/ libcrypto.so.1	• /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 • /opt/openssl/1.0/lib/hpux64/libcrypto.so.1

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

Table 5 OpenSSL A.01.00.02h PA-RISC libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/1.0/lib/libssl.1.0.2h.a	• /usr/lib/libssl.a * • /opt/openssl/lib/libssl.a * • /opt/openssl/0.9.8/lib/libssl.1.0.2h.a • /opt/openssl/0.9.7/lib/libssl.1.0.2h.a • /opt/openssl/1.0/lib/libssl.a
	/opt/openssl/1.0/lib/libcrypto.1.0.2h.a	• /usr/lib/libcrypto.a * • /opt/openssl/lib/libcrypto.a * • /opt/openssl/0.9.8/lib/libcrypto.1.0.2h.a • /opt/openssl/0.9.7/lib/libcrypto.1.0.2h.a • /opt/openssl/1.0/lib/libcrypto.a
32-bit shared	/opt/openssl/1.0/lib/libssl.sl.1.0.0	• /usr/lib/libssl.sl * • /usr/lib/libssl.sl.1.0.0 • /opt/openssl/lib/libssl.sl * • /opt/openssl/lib/libssl.sl.1.0.0 • /opt/openssl/0.9.8/lib/libssl.sl.1.0.0 • /opt/openssl/0.9.7/lib/libssl.sl.1.0.0 • /opt/openssl/1.0/lib/libssl.sl
	/opt/openssl/1.0/lib/libcrypto.sl.1.0.0	• /usr/lib/libcrypto.sl * • /usr/lib/libcrypto.sl.1.0.0 • /opt/openssl/lib/libcrypto.sl * • /opt/openssl/lib/libcrypto.sl.1.0.0 • /opt/openssl/0.9.8/lib/libcrypto.sl.1.0.0 • /opt/openssl/0.9.7/lib/libcrypto.sl.1.0.0 • /opt/openssl/1.0/lib/libcrypto.sl
64-bit static	/opt/openssl/1.0/lib/pa20_64/libssl.1.0.2h.a	• /usr/lib/pa20_64/libssl.a * • /opt/openssl/lib/pa20_64/libssl.a * • /opt/openssl/0.9.8/lib/pa20_64/libssl.1.0.2h.a • /opt/openssl/0.9.7/lib/pa20_64/libssl.1.0.2h.a • /opt/openssl/1.0/lib/pa20_64/libssl.a
	/opt/openssl/1.0/lib/pa20_64/libcrypto.1.0.2h.a	• /usr/lib/pa20_64/libcrypto.a * • /opt/openssl/lib/pa20_64/libcrypto.a * • /opt/openssl/0.9.8/lib/pa20_64/libcrypto.1.0.2h.a • /opt/openssl/0.9.7/lib/pa20_64/libcrypto.1.0.2h.a • /opt/openssl/1.0/lib/pa20_64/libcrypto.a

Table 5 OpenSSL A.01.00.02h PA-RISC libraries *(continued)*

Library	Library Name/Location	Symbolic Link
64-bit shared	/opt/openssl/1.0/lib/ pa20_64/ libssl.sl.1.0.0	• /usr/lib/pa20_64/libssl.sl * • /usr/lib/pa20_64/libssl.sl.1.0.0 • /opt/openssl/lib/pa20_64/libssl.sl * • /opt/openssl/lib/pa20_64/libssl.sl.1.0.0 • /opt/openssl/0.9.8/lib/pa20_64/ libssl.sl.1.0.0 • /opt/openssl/0.9.7/lib/pa20_64/ libssl.sl.1.0.0 • /opt/openssl/1.0/lib/pa20_64/libssl.sl
	/opt/openssl/1.0/lib/ pa20_64/ libcrypto.sl.1.0.0	• /usr/lib/pa20_64/libcrypto.sl * • /usr/lib/pa20_64/libcrypto.sl.1.0.0 • /opt/openssl/lib/pa20_64/libcrypto.sl * • /opt/openssl/lib/pa20_64/libcrypto.sl.1.0.0 • /opt/openssl/0.9.8/lib/pa20_64/ libcrypto.sl.1.0.0 • /opt/openssl/0.9.7/lib/pa20_64/ libcrypto.sl.1.0.0 • /opt/openssl/1.0/lib/pa20_64/libcrypto.sl

NOTE: Symbolic links marked * are applicable only if the default version is OpenSSL A.01.00.02h.

Table 6 OpenSSL A.01.00.02h Intel Itanium® libraries

Library	Library Name/Location	Symbolic Link
32-bit static	/opt/openssl/1.0/lib/ hpux32/ libssl.1.0.2h.a	• /usr/lib/hpux32/libssl.a * • /opt/openssl/lib/hpux32/libssl.a * • /opt/openssl/0.9.8/lib/hpux32/ libssl.1.0.2h.a • /opt/openssl/0.9.7/lib/hpux32/ libssl.1.0.2h.a • /opt/openssl/1.0/lib/hpux32/libssl.a
	/opt/openssl/1.0/lib/ hpux32/ libcrypto.1.0.2h.a	• /usr/lib/hpux32/libcrypto.a * • /opt/openssl/lib/hpux32/libcrypto.a * • /opt/openssl/0.9.8/lib/hpux32/ libcrypto.1.0.2h.a • /opt/openssl/0.9.7/lib/hpux32/ libcrypto.1.0.2h.a • /opt/openssl/1.0/lib/hpux32/libcrypto.a

Table 6 OpenSSL A.01.00.02h Intel Itanium® libraries *(continued)*

Library	Library Name/Location	Symbolic Link
32-bit shared	/opt/openssl/1.0/lib/ hpux32/ libssl.so.1.0.0	• /usr/lib/hpux32/libssl.so * • /usr/lib/hpux32/libssl.so.1.0.0 • /opt/openssl/lib/hpux32/libssl.so * • /opt/openssl/lib/hpux32/libssl.so.1.0.0 • /opt/openssl/0.9.8/lib/hpux32/ libssl.so.1.0.0 • /opt/openssl/0.9.7/lib/hpux32/ libssl.so.1.0.0 • /opt/openssl/1.0/lib/hpux32/libssl.so
	/opt/openssl/1.0/lib/ hpux32/ libcrypto.so.1.0.0	• /usr/lib/hpux32/libcrypto.so * • /usr/lib/hpux32/libcrypto.so.1.0.0 • /opt/openssl/lib/hpux32/libcrypto.so * • /opt/openssl/lib/hpux32/libcrypto.so.1.0.0 • /opt/openssl/0.9.8/lib/hpux32/ libcrypto.so.1.0.0 • /opt/openssl/0.9.7/lib/hpux32/ libcrypto.so.1.0.0 • /opt/openssl/1.0/lib/hpux32/libcrypto.so
64-bit static	/opt/openssl/1.0/lib/ hpux64/ libssl.1.0.2h.a	• /usr/lib/hpux64/libssl.a * • /opt/openssl/lib/hpux64/libssl.a * • /opt/openssl/0.9.8/lib/hpux64/ libssl.1.0.2h.a • /opt/openssl/0.9.7/lib/hpux64/ libssl.1.0.2h.a • /opt/openssl/1.0/lib/hpux64/libssl.a
	/opt/openssl/1.0/lib/ hpux64/ libcrypto.1.0.2h.a	• /usr/lib/hpux64/libcrypto.a * • /opt/openssl/lib/hpux64/libcrypto.a * • /opt/openssl/0.9.8/lib/hpux64/ libcrypto.1.0.2h.a • /opt/openssl/0.9.7/lib/hpux64/ libcrypto.1.0.2h.a • /opt/openssl/1.0/lib/hpux64/libcrypto.a

Table 6 OpenSSL A.01.00.02h Intel Itanium® libraries *(continued)*

Library	Library Name/Location	Symbolic Link
64-bit shared	/opt/openssl/1.0/lib/ hpux64/ libssl.so.1.0.0	• /usr/lib/hpux64/libssl.so * • /usr/lib/hpux64/libssl.so.1.0.0 • /opt/openssl/lib/hpux64/libssl.so * • /opt/openssl/lib/hpux64/libssl.so.1.0.0 • /opt/openssl/0.9.8/lib/hpux64/ libssl.so.1.0.0 • /opt/openssl/0.9.7/lib/hpux64/ libssl.so.1.0.0 • /opt/openssl/1.0/lib/hpux64/libssl.so
	/opt/openssl/1.0/lib/ hpux64/ libcrypto.so.1.0.0	• /usr/lib/hpux64/libcrypto.so * • /usr/lib/hpux64/libcrypto.so.1.0.0 • /opt/openssl/lib/hpux64/libcrypto.so * • /opt/openssl/lib/hpux64/libcrypto.so.1.0.0 • /opt/openssl/0.9.8/lib/hpux64/ libcrypto.so.1.0.0 • /opt/openssl/0.9.7/lib/hpux64/ libcrypto.so.1.0.0 • /opt/openssl/1.0/lib/hpux64/libcrypto.so

NOTE: Symbolic links marked * are applicable only if the default version is OpenSSL A.01.00.02h.

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

1.4.2 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 of files.
- Creating and viewing RSA, DSA, and DH public keys.
- Encrypting or decrypting files using public keys.
- Creating X.509 certificates, certificate requests, and Certificate Revocation Lists (CRL).
- Managing the Certificate Authority (CA).

1.4.3 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
```

1.5 Defects fixed in OpenSSL

This version includes several changes and fixes. For more information on the fixes, see the OpenSSL Changelog at **http://www.openssl.org/news/changelog.html** and OpenSSL Release Notes at **https://www.openssl.org/news/openssl-1.0.2-notes.html**.

1.6 Compatibility Information

- OpenSSL A.1.01.00.01j and later on HP-UX 11i v3 will use sha256 as the default_md and 2048 as default_bits in `openssl.cnf` file. OpenSSL 0.9.8zf still uses sha1 as default_md and 1024 as default_bits in the `openssl.cnf` file.
- This is the first release of OpenSSL 1.0.2 version on HP-UX 11i v3. These libraries are not backward compatible with the earlier version of OpenSSL 0.9.8 (and 0.9.7) libraries. Validate the applications that use OpenSSL, before using the depot in your production environment.
- Following are the known compatibility differences from OpenSSL version A.01.00.01s onwards:
  - By default, SSLv2 protocol is disabled.
  - By default, "EXPORT" or "LOW" strength ciphers in SSLv3 and later are disabled.

Before updating OpenSSL version A.01.00.01s and later, end-user must validate the above disabled protocol and ciphers in their environment/application. If any of the applications (or application using OpenSSL command line utility) are using the disabled protocol /ciphers, they might fail and needs to be changed before the migration.

For more information, see the OpenSSL advisory at **https://openssl.org/news/secadv/20160301.txt**.

1.7 Installation Requirements

This section lists the system and patch requirements for OpenSSL.

1.7.1 System requirements

Table 7 specifies the minimum system requirements for installing OpenSSL.

Table 7 System requirements for installing OpenSSL

Component	Requirement
Operating system	HP-UX 11i v3
Hardware	• PA-RISC • IA-64

Table 7 System requirements for installing OpenSSL *(continued)*

Component	Requirement
Disk space	650 MB
Software availability in native languages	English only

1.7.2 Patch requirements

HPE has tested the A.01.00.02h software in test environments with the Support Plus media listed in Table 8.

Table 8 Patch bundle required for HP-UX 11i v3 customers

Operating System	Required Support Plus Media
	Date
HP-UX 11i v3	March 2014 or later release

For more information about the Support Plus media, see HPE Support Center at: **http://www.hpe.com/support/hpesc**.

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.

1.8 Installing OpenSSL

To install OpenSSL, complete the following steps:

1. Log in as root.
2. Insert the software CD into the appropriate drive if you are installing from the Application Software 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. Run the following command:

   ```
   $swinstall -s  <fully qualified depot source path>
   ```

4. Enter the drive mount point in the **Source Depot Path** box and click **OK**. Change the **Source Host Name** if needed.
5. Select the OpenSSL bundle and enter the product sublevel.
6. Select the OpenSSL product from the list of available products and choose **Mark for Install** from the **Actions** menu.
7. Select **Install** from the **Actions** menu to begin installation.
8. Select **OK** in the **Install Analysis** window when the **Status** box displays a message to say it is ready.
9. Select **Yes** to begin the installation.

   NOTE: `swinstall` installs OpenSSL in /opt/openssl

The HP-UX OpenSSL files are loaded in approximately three to five minutes.

IMPORTANT: You cannot install OpenSSL 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 the current version of OpenSSL.

1.9 Using the Openssl command-line tool

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

1.9.1 Options

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

Table 9 The Openssl command-line options

Option Name	Description
`ca`	Certificate Authority management
`crl`	Certificate Revocation List management
`dgst`	Message digest calculation
`dsa`	DSA key management
`enc`	Encrypting files with ciphers
`gendsa`	Generation of DSA keys
`genrsa`	Generation of RSA keys
`req`	X.509 Certificate Request management
`rsa`	RSA key management
`verify`	X.509 certificate verification
`x509`	X.509 certificate management

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

1.9.2 Using Openssl

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

1.9.2.1 Creating RSA keys

The 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 <filename> <bits>`

Where:

<bits> Specifies the size of the key.

<filename> Specifies the file name for storing the key pair.

For example: `# openssl genrsa -out key.pem 1024`

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

1.9.2.2 Creating a password-protected RSA key pair

The 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).
<filename>	specifies the file name for storing the key pair.
<bits>	specifies the key size.

For example: **# openssl genrsa -des3 -out key.pem 1024**.

This command creates a 3DES-encrypted 1024-bit key pair stored in the file `key.txt`. The encryption is done using the pass phrase supplied by the user.

1.9.2.3 Viewing an RSA key pair

The following is the syntax to view an RSA key pair:

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

For example: **# openssl genrsa -des3 -out key.pem 1024**.

This command displays the modulus, exponent, and prime key values of the key pair stored in the `key.pem` file. If the key pair stored in `key.pem` is encrypted, then these commands prompt the user for the pass phrase.

1.9.2.4 Creating an RSA certificate request

The 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.

For example: **# openssl req -new -nodes -out cert.txt -keyout key.pem -subj "/C=US/ST=CA/L=CITY/CN=localhost/emailAddress=root@localhost"**

This command creates an RSA certificate request.

1.9.2.5 Creating a self-signed certificate

The 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.

For example: **# 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"**

This command creates a self-signed certificate.

1.10 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.

1.10.1 Getting the OpenSSL software

You can obtain OpenSSL A.01.00.02h software from the following sources:

- HPE Software Depot at: **https://www.hpe.com/support/softwaredepot.**
- Application Software CDs
- HP-UX Operating Environments (OEs)

1.10.2 Learning about OpenSSL technology

A large volume of information exists on the Internet about OpenSSL technology. HPE 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: **https://www.openssl.org/community/mailinglists.html**

OpenSSL A.01.00.02h.001 Release Notes is available at the following locations:

- The PDF version is available at: **HPE Support Center Product Page**.
- A text version of the `README.hp` readme file in the `/opt/openssl` directory.
- A text version of the FIPSs `README.hp` readme file in the `/opt/openssl/fips/1.0` directory.

2 Frequently asked questions (FAQs)

The following are questions frequently asked about OpenSSL.

2.1 What does OpenSSL do? Why do I need it?

OpenSSL offers an advanced level of security using the SSL or TLS protocols. Client-server applications that send and receive data over a network are open to a range of vulnerabilities. They can use SSL or TLS to implement privacy (through encryption), tamper-proofing (through message digests), and non-repudiation (through certificates and digital signatures).

2.2 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.

2.3 There are several flavors of libraries available in OpenSSL. What are they? How do I know when to use which library?

Use the OpenSSL 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 11). You can also choose to create user applications using either a static library or a shared library. In addition, OpenSSL contains libraries that support hardware ENGINES.

2.4 How do I switch between OpenSSL A.00.09.07m, A.00.09.08zf, and OpenSSL A.01.00.02h?

During installation, the depot installs OpenSSL A.00.09.07m, A.00.09.08zf, and OpenSSL A.01.00.02h. in the `/opt/openssl/0.9.7`, `/opt/openssl/0.9.8`, and `/opt/openssl/1.0` 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.

2.5 How does the performance of OpenSSL A.00.09.07m, A.00.09.08zf, or OpenSSL A.01.00.02h compare to the Open Source version 0.9.7m, 0.9.8zf, or 1.0.2h 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://www.hpe.com/support/softwaredepot.

2.6 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.

2.7 How can I install OpenSSL A.01.00.02h?

Depending on the method of installation from the application CD or the web, follow the installation instructions.

2.8 How can I uninstall OpenSSL A.01.00.02h?

Use the following command to uninstall OpenSSL:

```
# swremove OpenSSL
```

NOTE: Multiple products are dependent on OpenSSL, HPE recommends not to uninstall the OpenSSL product.

2.9 I have already got version A.00.09.08y on my HP-UX system, and I am quite happy with it. Why do I need to move to OpenSSL A.01.00.02h?

This new version of OpenSSL contains security fixes apart from other bug fixes and enhancements. These critical fixes are well publicized at the OpenSSL site. HPE recommends that you upgrade to OpenSSL A.01.00.02h even if you are not affected by these defects.

2.10 If I want my existing applications to use this new version of OpenSSL, do I need to rebuild the application? Do you guarantee binary compatibility of my applications with this new version?

OpenSSL A.01.00.02h contains a precompiled version of OpenSSL A.00.09.07m and OpenSSL A.00.09.08zf. Hence, applications linked directly with A.00.09.07 and A.00.09.08 versions of OpenSSL libssl, libcrypto shared libraries will continue to work with some exceptions.

You can check the version of OpenSSL library that the application is directly linked against using the `chatr` command on all the executable and shared libraries of the application.

```
libcrypto.sl.1.0.0/libcrypto.so.1.0.0— OpenSSL 1.0 crypto library
libssl.sl.1.0.0/libssl.so.1.0.0— OpenSSL 1.0 SSL library
libcrypto.sl.1/libcrypto.so.1— OpenSSL 0.9.8 cypto library
libssl.sl.1/libssl.so.1— OpenSSL 0.9.8 SSL library
libcrypto.sl.0/libcrypto.so.0— OpenSSL 0.9.7 crypto library
libssl.sl.0/libssl.so.0— OpenSSL 0.9.7 SSL library
```

If the application is linked against the above libraries, the application continues to work with some exceptions. However, HPE recommends to rebuild the application with the latest version of OpenSSL libraries and headers to ensure that the application runs with the latest security fixes.

There are chances that the application is built against OpenSSL not provided by HPE. The `chatr` command output of the application executable and shared libraries shows the following OpenSSL libraries as linked to the application:

- libcrypto.sl
- libcrypto.so
- libssl.sl
- libssl.so

On the HP-UX system, these above files are soft links which can point to any of the OpenSSL 0.9.7/0.9.8 or 1.0 libraries. The current version of OpenSSL A.01.00.02h will by default point these soft links to OpenSSL 1.0 libraries. Applications might not work as expected, since libcrypto.sl/libcrypto.so and libssl.sl/libssl.so will point to OpenSSL 1.0 libraries. HPE recommends to rebuild the application against the OpenSSL A.01.00.02h header files and libraries provided by HPE.

In case applications are dynamically loading (dlopen())/shl_load()) the OpenSSL libcrypto.sl/libcrypto.so or libssl.sl/libssl.so libraries (instead of dynamically loading specific libcrypto.sl.0/libcrypto.so.0, libcrypto.sl.1/libcrypto.so.1, libcrypto.sl.1.0.0/libcrypto.so.1.0.0, libssl.sl.0/libssl.so.0, libssl.sl.1/libssl.so.1 or libssl.sl.1.0.0/libssl.so.1.0.0 libraries) then it is required to rebuild the application based on the version of OpenSSL you are moving to.

If the application at runtime loads different incompatible versions of OpenSSL libraries (for example, OpenSSL 0.9.8 and OpenSSL 1.0 libraries), either due to direct linking by the application or indirect linking via dependent shared libraries, the behavior of the application is unknown.

The OpenSSL 1.0, 0.9.8, and 0.9.7 libraries are not binary compatible with each other.

The OpenSSL 1.0 libraries are binary compatible with the minor releases of OpenSSL 1.0 libraries (1.0.2h, 1.0.1m, 1.0.1p and so on).

The OpenSSL 0.9.8 libraries is binary compatible with the minor releases of OpenSSL 0.9.8 libraries (0.9.8zf, 0.9.8x, 0.9.8w, and so on).

The OpenSSL 0.9.7 libraries is binary compatible with the minor releases of OpenSSL 0.9.7 libraries (0.9.7m, 0.9.7d, 0.9.7e, 0.9.7l, and so on).

2.11 I have HP-UX Internet Express OpenSSL 0.9.7c installed on my system. Will installing OpenSSL A.01.00.02h automatically uninstall HP-UX Internet Express OpenSSL 0.9.7c?

No. The HP-UX Internet Express OpenSSL 0.9.7c and A.01.00.02h 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.01.00.02h, you must first uninstall the HP-UX Internet Express OpenSSL product. You can use the `swremove ixOpenSSL` command to remove the product. The installation of the OpenSSL A.01.00.02h fails if the HP-UX Internet Express OpenSSL 0.9.7c product is present on the system.

2.12 I have already built Open Source OpenSSL 1.0.2h by downloading the source code directly from **http://www.openssl.org**. Now, I want to upgrade to OpenSSL A.01.00.02h. What must I do? Do I have to remove the pre-existing OpenSSL product from my system?

You may have a conflict depending on the location of OpenSSL 1.0.2h on your system. HPE recommends that you uninstall the previous OpenSSL version before installing OpenSSL A.01.00.02h.

2.13 Will HPE support recompiled versions of OpenSSL A.01.00.02h?

HPE does not support recompiled versions of OpenSSL A.01.00.02h. The source code is provided only for reference.

2.14 Why are idea, rc5, and mdc2 algorithms not configured in OpenSSL A.01.00.02h?

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

2.15 How do I find out which OpenSSL version I have on my system? What if I have more than one version?

Use the `#swlist` command to find out which version of OpenSSL is running on your system. You can also use the `what` command to verify the OpenSSL version number on your machine. It is not possible to have both the InternetExpress version and a supported HP-UX OpenSSL version of OpenSSL on the same machine. The following are examples of using the `#swlist` and `what` commands.

Example 1 You have the Internet Express version installed on your machine.

```
# swlist | grep -i openssl
ixOpenSSL A.02.00-0.9.7c  Secure Network Communications Protocol

# what openssl
OpenSSL A.02.00-0.9.7c
```

Example 2 You have the OpenSSL A.00.09.07i version installed on your machine.

```
# swlist | grep -i openssl
OpenSSL A.00.09.07i.005 Secure Network Communications Protocol

# what openssl
OpenSSL A.00.09.07i.005
```

Example 3 You do not have an HP-UX depot installed, but have downloaded the source code and built the product yourself.

```
# swlist | grep -i openssl
This will not return anything, since you do not have an OpenSSL depot
installed on your machine.

# what openssl
 Unless you included $what strings into the source code before building the product, this command will
 not display any results either.
```

2.16 A lot of information is available for OpenSSL-enabler products, such as Stunnel. How is the procedure for installation in Stunnel different from the OpenSSL installation?

Consider a client-server application sending and receiving unencrypted data over the network. You now want to achieve a higher level of security for your application. The following methods describe how to use OpenSSL technology to encrypt client or server 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 a Stunnel-based environment in which data from the client application is sent to a 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.

2.17 I cannot find the Verisign root certificates in `/opt/openssl/certs` directory. Where are the root certificates?

All the root certificates are not shipped by default in `/opt/openssl/certs` directory. You can get the VeriSign Root certificates from **http://www.symantec.com/page.jsp?id=roots**.

2.18 What does FIPS stand for?

Federal Information Processing Standards.

2.19 Where can I find FIPS related material?

See Openssl official website **http://www.openssl.org/docs/fips**.

2.20 How can I use openssl executable in FIPS mode?

Set environment variable OPENSSL_FIPS to 1 (export OPENSSL_FIPS=1).

2.21 How do I enter FIPS mode in application?

After the application calls FIPS_mode_set() successfully, it will be using FIPS 140-2 Approved mode of OpenSSL when it links with `libcrypto.so.1.0.0`, `libssl.so.1.0.0`, `libcryto.a`, and `libssl.a` provided in `/opt/openssl/fips/1.0/lib/hpux32` or `/opt/openssl/fips/1.0/lib/hpux64` directory.

In case, the application is linking against the dynamic version of OpenSSL libraries (`libcrypto.so.1.0.0`, `libssl.so.1.0.0`), ensure that the application is only using the libraries from `/opt/openssl/fips/1.0/lib/hpux32` or `/opt/openssl/fips/1.0/lib/hpux64` directory at run time.

One way of achieving this is to explicitly set the search path by using the "+b" linker directive (See ld_ia(1)). For example,

cc -Wl,+nodefaultrpath -Wl,+b /opt/openssl/fips/1.0/lib/hpux32 -lcrypto -lssl application1.c

Other options are to set the LD_LIBRARY_PATH or SHLIB_PATH environment variable, to point to the `/opt/openssl/fips/1.0/lib/hpux32` or `/opt/openssl/fips/1.0/lib/hpux64` directory before running the application.

2.22 Is there a sample application which uses FIPS capable openssl libraries?

See `/opt/openssl/fips/1.0/README.hp` for a sample application which uses FIPS capable openssl libraries.

2.23 What kind of algorithms I can use in FIPS mode?

To know the complete list of algorithms supported in FIPS approved mode, see *OpenSSL FIPS 140-2 Security Policy Version 2.0.X* at **http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1747.pdf**.

2.24 What happens when I use FIPS unallowed algorithms in FIPS mode?

The OpenSSL 140-2 User Guide mentions "the OpenSSL API attempts to disable non-FIPS algorithms, when in FIPS mode, at the EVP level and via most low level function calls. Failure to check the return code from low level functions could result in unexpected behavior. Note also that sufficiently creative or unusual use of the API may still allow the use of non-FIPS algorithms. The non-FIPS algorithm disabling is intended as an aid to the developer in preventing the accidental use of non-FIPS algorithms in FIPS mode, and not as an absolute guarantee."

In most cases, some messages like the following will appear.

- md5_dgst.c(74): OpenSSL internal error, assertion failed: Low level API call to digest MD5 forbidden in FIPS mode! ABORT instruction (core dumped)
- fips_enc.c(87): OpenSSL internal error, assertion failed: Cipher init previous FIPS forbidden algorithm error ignored ABORT instruction (core dumped)
- Error setting digest MD5

  2628:error:0608008D:digital envelope routines:EVP_DigestInit:disabled for fips:digest.c:237

2.25 What is the FIPS relationship to the OpenSSL API?

The "FIPS Object Module" is the special monolithic object module built from the special source distribution identified in the OpenSSL Security Policy. See **https://www.openssl.org/docs/fips/**. 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. When the "FIPS Object Module" and a FIPS compatible OpenSSL are separately built and installed on a system, the combination is called as a FIPS capable OpenSSL.

2.26 Why are idea, rc5 and mdc2 not configured in FIPS-OPENSSL-2.0.5?

Crypto algorithms idea, rc5, and mdc2 have patent issues. HPE will not redistribute any software using those algorithms.

2.27 I built my application using latest OpenSSL libraries. During runtime some of the crypto api's are not working as expected?

Change the order in which OpenSSL libraries are linked to your application and check the behavior.

3 Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.

3.1 More on OpenSSL documentation

For more information on documentation and other manuals of HP-UX OpenSSL Software, see **HP-UX OpenSSL Software**.

3.2 Support policies for HP-UX

For more information about support policy of HP-UX, see **HP-UX Support Policy**.