NAME
openssl – OpenSSL command line tool

SYNOPSIS
openssl command [ command_opts ] [ command_args ]
openssl [ list-standard-commands | list-message-digest-commands | list-cipher-commands | list-cipher-algorithms | list-message-digest-algorithms | list-public-key-algorithms]
openssl no−XXX [ arbitrary options ]

DESCRIPTION
OpenSSL is a cryptography toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols and related cryptography standards required by them.

The openssl program is a command line tool for using the various cryptography functions of OpenSSL’s crypto library from the shell. It can be used for

- Creation and management of private keys, public keys and parameters
- Public key cryptographic operations
- Creation of X.509 certificates, CSRs and CRLs
- Calculation of Message Digests
- Encryption and Decryption with Ciphers
- SSL/TLS Client and Server Tests
- Handling of S/MIME signed or encrypted mail
- Time Stamp requests, generation and verification

COMMAND SUMMARY
The openssl program provides a rich variety of commands (command in the SYNOPSIS above), each of which often has a wealth of options and arguments (command_opts and command_args in the SYNOPSIS).

The pseudo-commands list-standard-commands, list-message-digest-commands, and list-cipher-commands output a list (one entry per line) of the names of all standard commands, message digest commands, or cipher commands, respectively, that are available in the present openssl utility.

The pseudo-commands list-cipher-algorithms and list-message-digest-algorithms list all cipher and message digest names, one entry per line. Aliases are listed as:

from => to

The pseudo-command list-public-key-algorithms lists all supported public key algorithms.

The pseudo-command no−XXX tests whether a command of the specified name is available. If no command named XXX exists, it returns 0 (success) and prints no−XXX; otherwise it returns 1 and prints XXX. In both cases, the output goes to stdout and nothing is printed to stderr. Additional command line arguments are always ignored. Since for each cipher there is a command of the same name, this provides an easy way for shell scripts to test for the availability of ciphers in the openssl program. (no−XXX is not able to detect pseudo-commands such as quit, list−...−commands, or no−XXX itself.)

STANDARD COMMANDS
asn1parse Parse an ASN.1 sequence.
ca Certificate Authority (CA) Management.
ciphers Cipher Suite Description Determination.
cms CMS (Cryptographic Message Syntax) utility
crl2pkcs7 CRL to PKCS#7 Conversion.
dgst Message Digest Calculation.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dhparam</td>
<td>Generation and Management of Diffie-Hellman Parameters. Superseded by <code>genpkey</code> and <code>pkeyparam</code></td>
</tr>
<tr>
<td>dsa</td>
<td>DSA Data Management.</td>
</tr>
<tr>
<td>dsaparam</td>
<td>DSA Parameter Generation and Management. Superseded by <code>genpkey</code> and <code>pkeyparam</code></td>
</tr>
<tr>
<td>ec</td>
<td>EC (Elliptic curve) key processing</td>
</tr>
<tr>
<td>ecparam</td>
<td>EC parameter manipulation and generation</td>
</tr>
<tr>
<td>engine</td>
<td>Engine (loadable module) information and manipulation.</td>
</tr>
<tr>
<td>errstr</td>
<td>Error Number to Error String Conversion.</td>
</tr>
<tr>
<td>gendsa</td>
<td>Generation of DSA Private Key from Parameters. Superseded by <code>genpkey</code> and <code>pkey</code></td>
</tr>
<tr>
<td>genpkey</td>
<td>Generation of Private Key or Parameters.</td>
</tr>
<tr>
<td>genrsa</td>
<td>Generation of RSA Private Key. Superceded by <code>genpkey</code>.</td>
</tr>
<tr>
<td>nseq</td>
<td>Create or examine a netscape certificate sequence</td>
</tr>
<tr>
<td>ocsp</td>
<td>Online Certificate Status Protocol utility.</td>
</tr>
<tr>
<td>passwd</td>
<td>Generation of hashed passwords.</td>
</tr>
<tr>
<td>pkcs12</td>
<td>PKCS#12 Data Management.</td>
</tr>
<tr>
<td>pkcs7</td>
<td>PKCS#7 Data Management.</td>
</tr>
<tr>
<td>pkey</td>
<td>Public and private key management.</td>
</tr>
<tr>
<td>pkeyparam</td>
<td>Public key algorithm parameter management.</td>
</tr>
<tr>
<td>pkeyutl</td>
<td>Public key algorithm cryptographic operation utility.</td>
</tr>
<tr>
<td>rand</td>
<td>Generate pseudo-random bytes.</td>
</tr>
<tr>
<td>req</td>
<td>PKCS#10 X.509 Certificate Signing Request (CSR) Management.</td>
</tr>
<tr>
<td>rsa</td>
<td>RSA key management.</td>
</tr>
<tr>
<td>rsautl</td>
<td>RSA utility for signing, verification, encryption, and decryption. Superseded by <code>pkeyutl</code></td>
</tr>
<tr>
<td>s_client</td>
<td>This implements a generic SSL/TLS client which can establish a transparent connection to a remote server speaking SSL/TLS. It’s intended for testing purposes only and provides only rudimentary interface functionality but internally uses mostly all functionality of the OpenSSL <code>ssl</code> library.</td>
</tr>
<tr>
<td>s_server</td>
<td>This implements a generic SSL/TLS server which accepts connections from remote clients speaking SSL/TLS. It’s intended for testing purposes only and provides only rudimentary interface functionality but internally uses mostly all functionality of the OpenSSL <code>ssl</code> library. It provides both an own command line oriented protocol for testing SSL functions and a simple HTTP response facility to emulate an SSL/TLS–aware webserver.</td>
</tr>
<tr>
<td>s_time</td>
<td>SSL Connection Timer.</td>
</tr>
<tr>
<td>sess_id</td>
<td>SSL Session Data Management.</td>
</tr>
<tr>
<td>smime</td>
<td>S/MIME mail processing.</td>
</tr>
<tr>
<td>speed</td>
<td>Algorithm Speed Measurement.</td>
</tr>
<tr>
<td>spkac</td>
<td>SPKAC printing and generating utility</td>
</tr>
<tr>
<td>ts</td>
<td>Time Stamping Authority tool (client/server)</td>
</tr>
</tbody>
</table>
verify        X.509 Certificate Verification.
version       OpenSSL Version Information.
x509          X.509 Certificate Data Management.

MESSAGE DIGEST COMMANDS
md2           MD2 Digest
md5           MD5 Digest
mdc2          MDC2 Digest
rmd160        RMD−160 Digest
sha           SHA Digest
sha1          SHA−1 Digest
sha224        SHA−224 Digest
sha256        SHA−256 Digest
sha384        SHA−384 Digest
sha512        SHA−512 Digest

ENCODING AND CIPHER COMMANDS
base64        Base64 Encoding
bf bf-cbc bf-cfb bf-ecb bf-ofb
Blowfish Cipher
cast cast-cbc
CAST Cipher
cast5−cbc cast5−cfb cast5−ecb cast5−ofb
CAST5 Cipher
des des-cbc des-cfb des-ecb des-ede-cbc des-ede-cfb des-ede-ofb des-ofb
DES Cipher
des3 desx des−ede3 des−ede3−cbc des−ede3−cfb des−ede3−ofb
Triple-DES Cipher
idea idea-cbc idea-cfb idea-ecb idea-ofb
IDEA Cipher
rc2 rc2−cbc rc2−cfb rc2−ecb rc2−ofb
RC2 Cipher
rc4           RC4 Cipher
rc5 rc5−cbc rc5−cfb rc5−ecb rc5−ofb
RC5 Cipher

PASS PHRASE ARGUMENTS
Several commands accept password arguments, typically using −passin and −passout for input and output passwords respectively. These allow the password to be obtained from a variety of sources. Both of these options take a single argument whose format is described below. If no password argument is given and a password is required then the user is prompted to enter one: this will typically be read from the current terminal with echoing turned off.

pass:password
the actual password is password. Since the password is visible to utilities (like 'ps' under Unix) this form should only be used where security is not important.

env:var
obtain the password from the environment variable var. Since the environment of other processes is visible on certain platforms (e.g. ps under certain Unix OSes) this option should be
used with caution.

file:pathname
the first line of pathname is the password. If the same pathname argument is supplied to
−passin and −passout arguments then the first line will be used for the input password and the
next line for the output password. pathname need not refer to a regular file: it could for
example refer to a device or named pipe.

fd:number
read the password from the file descriptor number. This can be used to send the data via a pipe
for example.

stdin read the password from standard input.

SEE ALSO
asn1parse (1), ca (1), config (5), crl (1), crl2pkcs7 (1), dgst (1), dhparam (1), dsa (1), dsaparam (1), enc (1),
gendsa (1), genpkey (1), genrsa (1), nseq (1), openssl (1), passwd (1), pkcs12 (1), pkcs7 (1), pkcs8 (1),
rand (1), req (1), rsa (1), rsautl (1), s_client (1), s_server (1), s_time (1), smime (1), spkac (1), verify (1),
version (1), x509 (1), crypto (3), ssl (3), x509v3_config (5)

HISTORY
The openssl (1) document appeared in OpenSSL 0.9.2. The list−XXX−commands pseudo-commands were
added in OpenSSL 0.9.3; The list−XXX−algorithms pseudo-commands were added in OpenSSL 1.0.0; the
no−XXX pseudo-commands were added in OpenSSL 0.9.5a. For notes on the availability of other
commands, see their individual manual pages.