ciphers(1)

NAME

ciphers - SSL cipher display and cipher list tool.

SYNOPSIS

openssl ciphers [**-v**] [**-V**] [**-ssl2**] [**-ssl3**] [**-tls1**] [**cipherlist**]

DESCRIPTION

The **ciphers** command converts textual OpenSSL cipher lists into ordered SSL cipher preference lists. It can be used as a test tool to determine the appropriate cipherlist.

COMMAND OPTIONS

-v

Verbose option. List ciphers with a complete description of protocol version (SSLv2 or SSLv3; the latter includes TLS), key exchange, authentication, encryption and mac algorithms used along with any key size restrictions and whether the algorithm is classed as an ``export'' cipher. Note that without the **-v** option, ciphers may seem to appear twice in a cipher list; this is when similar ciphers are available for SSL v2 and for SSL v3/TLS v1.

-V

Like **-V**, but include cipher suite codes in output (hex format).

-ssl3

only include SSL v3 ciphers.

-ssl2

only include SSL v2 ciphers.

-tls1

only include TLS v1 ciphers.

-h, -?

print a brief usage message.

cipherlist

a cipher list to convert to a cipher preference list. If it is not included then the default cipher list will be used. The format is described below.

CIPHER LIST FORMAT

The cipher list consists of one or more *cipher strings* separated by colons. Commas or spaces are also acceptable separators but colons are normally used.

The actual cipher string can take several different forms.

It can consist of a single cipher suite such as **RC4-SHA**.

It can represent a list of cipher suites containing a certain algorithm, or cipher suites of a certain type. For example **SHA1** represents all ciphers suites using the digest algorithm SHA1 and **SSLv3** represents all SSL v3 algorithms.

Lists of cipher suites can be combined in a single cipher string using the **+** character. This is used as a logical **and** operation. For example **SHA1+DES** represents all cipher suites containing the SHA1 **and** the DES algorithms.

Each cipher string can be optionally preceded by the characters **!**, **-** or **+**.

If **!** is used then the ciphers are permanently deleted from the list. The ciphers deleted can never reappear in the list even if they are explicitly stated.

If **-** is used then the ciphers are deleted from the list, but some or all of the ciphers can be added again by later options.

If **+** is used then the ciphers are moved to the end of the list. This option doesn't add any new ciphers it just moves matching existing ones.

If none of these characters is present then the string is just interpreted as a list of ciphers to be appended to the current preference list. If the list includes any ciphers already present they will be ignored: that is they will not moved to the end of the list.

Additionally the cipher string **@STRENGTH** can be used at any point to sort the current cipher list in order of encryption algorithm key length.

CIPHER STRINGS

The following is a list of all permitted cipher strings and their meanings.

DEFAULT

the default cipher list. This is determined at compile time and, as of OpenSSL 1.0.0, is normally **ALL:!aNULL:!eNULL**. This must be the first cipher string specified.

COMPLEMENTOFDEFAULT

the ciphers included in **ALL**, but not enabled by default. Currently this is **ADH**. Note that this rule does not cover **eNULL**, which is not included by **ALL** (use **COMPLEMENTOFALL** if necessary).

ALL

all cipher suites except the **eNULL** ciphers which must be explicitly enabled; as of OpenSSL, the **ALL** cipher suites are reasonably ordered by default

COMPLEMENTOFALL

the cipher suites not enabled by **ALL**, currently being **eNULL**.

HIGH

``high'' encryption cipher suites. This currently means those with key lengths larger than 128 bits, and some cipher suites with 128-bit keys.

MEDIUM

``medium'' encryption cipher suites, currently some of those using 128 bit encryption.

LOW

``low'' encryption cipher suites, currently those using 64 or 56 bit encryption algorithms but excluding export cipher suites.

EXP, EXPORT

export encryption algorithms. Including 40 and 56 bits algorithms.

EXPORT40

40 bit export encryption algorithms

EXPORT56

56 bit export encryption algorithms. In OpenSSL 0.9.8c and later the set of 56 bit export ciphers is empty unless OpenSSL has been explicitly configured with support for experimental ciphers.

eNULL, NULL

the ``NULL'' ciphers that is those offering no encryption. Because these offer no encryption at all and are a security risk they are disabled unless explicitly included.

aNULL

the cipher suites offering no authentication. This is currently the anonymous DH algorithms. These cipher suites are vulnerable to a ``man in the middle'' attack and so their use is normally discouraged.

kRSA, RSA

cipher suites using RSA key exchange.

kEDH

cipher suites using ephemeral DH key agreement.

kDHr, kDHd

cipher suites using DH key agreement and DH certificates signed by CAs with RSA and DSS keys respectively. Not implemented.

aRSA

cipher suites using RSA authentication, i.e. the certificates carry RSA keys.

aDSS, DSS

cipher suites using DSS authentication, i.e. the certificates carry DSS keys.

aDH

cipher suites effectively using DH authentication, i.e. the certificates carry DH keys. Not implemented.

kFZA, aFZA, eFZA, FZA

ciphers suites using FORTEZZA key exchange, authentication, encryption or all FORTEZZA algorithms. Not implemented.

TLSv1, SSLv3, SSLv2

TLS v1.0, SSL v3.0 or SSL v2.0 cipher suites respectively.

DH

cipher suites using DH, including anonymous DH.

ADH

anonymous DH cipher suites.

AES

cipher suites using AES.

CAMELLIA

cipher suites using Camellia.

3DES

cipher suites using triple DES.

DES

cipher suites using DES (not triple DES).

RC4

cipher suites using RC4.

RC2

cipher suites using RC2.

IDEA

cipher suites using IDEA.

SEED

cipher suites using SEED.

MD5

cipher suites using MD5.

SHA1, SHA

cipher suites using SHA1.

aGOST

cipher suites using GOST R 34.10 (either 2001 or 94) for authenticaction (needs an engine supporting GOST algorithms).

aGOST01

cipher suites using GOST R 34.10-2001 authentication.

aGOST94

cipher suites using GOST R 34.10-94 authentication (note that R 34.10-94 standard has been expired so use GOST R 34.10-2001)

kGOST

cipher suites, using VKO 34.10 key exchange, specified in the RFC 4357.

GOST94

cipher suites, using HMAC based on GOST R 34.11-94.

GOST89MAC

cipher suites using GOST 28147-89 MAC **instead of** HMAC.

CIPHER SUITE NAMES

The following lists give the SSL or TLS cipher suites names from the relevant specification and their OpenSSL equivalents. It should be noted, that several cipher suite names do not include the authentication used, e.g. DES-CBC3-SHA. In these cases, RSA authentication is used.

SSL v3.0 cipher suites.

```
 SSL_RSA_WITH_NULL_MD5                   NULL-MD5
 SSL_RSA_WITH_NULL_SHA                   NULL-SHA
 SSL_RSA_EXPORT_WITH_RC4_40_MD5          EXP-RC4-MD5
 SSL_RSA_WITH_RC4_128_MD5                RC4-MD5
 SSL_RSA_WITH_RC4_128_SHA                RC4-SHA
 SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5      EXP-RC2-CBC-MD5
 SSL_RSA_WITH_IDEA_CBC_SHA               IDEA-CBC-SHA
 SSL_RSA_EXPORT_WITH_DES40_CBC_SHA       EXP-DES-CBC-SHA
 SSL_RSA_WITH_DES_CBC_SHA                DES-CBC-SHA
 SSL_RSA_WITH_3DES_EDE_CBC_SHA           DES-CBC3-SHA

 SSL_DH_DSS_EXPORT_WITH_DES40_CBC_SHA    Not implemented.
 SSL_DH_DSS_WITH_DES_CBC_SHA             Not implemented.
 SSL_DH_DSS_WITH_3DES_EDE_CBC_SHA        Not implemented.
 SSL_DH_RSA_EXPORT_WITH_DES40_CBC_SHA    Not implemented.
 SSL_DH_RSA_WITH_DES_CBC_SHA             Not implemented.
 SSL_DH_RSA_WITH_3DES_EDE_CBC_SHA        Not implemented.
 SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA   EXP-EDH-DSS-DES-CBC-SHA
 SSL_DHE_DSS_WITH_DES_CBC_SHA            EDH-DSS-CBC-SHA
 SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA       EDH-DSS-DES-CBC3-SHA
 SSL_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA   EXP-EDH-RSA-DES-CBC-SHA
 SSL_DHE_RSA_WITH_DES_CBC_SHA            EDH-RSA-DES-CBC-SHA
 SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA       EDH-RSA-DES-CBC3-SHA

 SSL_DH_anon_EXPORT_WITH_RC4_40_MD5      EXP-ADH-RC4-MD5
 SSL_DH_anon_WITH_RC4_128_MD5            ADH-RC4-MD5
 SSL_DH_anon_EXPORT_WITH_DES40_CBC_SHA   EXP-ADH-DES-CBC-SHA
 SSL_DH_anon_WITH_DES_CBC_SHA            ADH-DES-CBC-SHA
 SSL_DH_anon_WITH_3DES_EDE_CBC_SHA       ADH-DES-CBC3-SHA

 SSL_FORTEZZA_KEA_WITH_NULL_SHA          Not implemented.
 SSL_FORTEZZA_KEA_WITH_FORTEZZA_CBC_SHA  Not implemented.
 SSL_FORTEZZA_KEA_WITH_RC4_128_SHA       Not implemented.
```

TLS v1.0 cipher suites.

TLS v1.0 cipher suites.

TLS_RSA_WITH_NULL_MD5	NULL-MD5
TLS_RSA_WITH_NULL_SHA	NULL-SHA
TLS_RSA_EXPORT_WITH_RC4_40_MD5	EXP-RC4-MD5
TLS_RSA_WITH_RC4_128_MD5	RC4-MD5
TLS_RSA_WITH_RC4_128_SHA	RC4-SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5	EXP-RC2-CBC-MD5
TLS_RSA_WITH_IDEA_CBC_SHA	IDEA-CBC-SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-DES-CBC-SHA
TLS_RSA_WITH_DES_CBC_SHA	DES-CBC-SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA	DES-CBC3-SHA
TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA	Not implemented.
TLS_DH_DSS_WITH_DES_CBC_SHA	Not implemented.
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	Not implemented.
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_DES_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	Not implemented.
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-DSS-DES-CBC-SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA	EDH-DSS-CBC-SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	EDH-DSS-DES-CBC3-SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA	EXP-EDH-RSA-DES-CBC-SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA	EDH-RSA-DES-CBC-SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	EDH-RSA-DES-CBC3-SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5	EXP-ADH-RC4-MD5
TLS_DH_anon_WITH_RC4_128_MD5	ADH-RC4-MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA	EXP-ADH-DES-CBC-SHA
TLS_DH_anon_WITH_DES_CBC_SHA	ADH-DES-CBC-SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA	ADH-DES-CBC3-SHA

AES ciphersuites from RFC3268, extending TLS v1.0

TLS_RSA_WITH_AES_128_CBC_SHA	AES128-SHA
TLS_RSA_WITH_AES_256_CBC_SHA	AES256-SHA
TLS_DH_DSS_WITH_AES_128_CBC_SHA	Not implemented.
TLS_DH_DSS_WITH_AES_256_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_AES_128_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_AES_256_CBC_SHA	Not implemented.
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE-DSS-AES128-SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE-DSS-AES256-SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE-RSA-AES128-SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE-RSA-AES256-SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA	ADH-AES128-SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA	ADH-AES256-SHA

Camellia ciphersuites from RFC4132, extending TLS v1.0

TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	CAMELLIA128-SHA
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	CAMELLIA256-SHA
TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA	Not implemented.
TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA	Not implemented.
TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA	DHE-DSS-CAMELLIA128-SHA
TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA	DHE-DSS-CAMELLIA256-SHA

TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA	DHE-RSA-CAMELLIA128-SHA
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA	DHE-RSA-CAMELLIA256-SHA
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA	ADH-CAMELLIA128-SHA
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA	ADH-CAMELLIA256-SHA

SEED ciphersuites from RFC4162, extending TLS v1.0

TLS_RSA_WITH_SEED_CBC_SHA	SEED-SHA
TLS_DH_DSS_WITH_SEED_CBC_SHA	Not implemented.
TLS_DH_RSA_WITH_SEED_CBC_SHA	Not implemented.
TLS_DHE_DSS_WITH_SEED_CBC_SHA	DHE-DSS-SEED-SHA
TLS_DHE_RSA_WITH_SEED_CBC_SHA	DHE-RSA-SEED-SHA
TLS_DH_anon_WITH_SEED_CBC_SHA	ADH-SEED-SHA

GOST ciphersuites from draft-chudov-cryptopro-cptls, extending TLS v1.0

Note: these ciphers require an engine which including GOST cryptographic algorithms, such as the **ccgost** engine, included in the OpenSSL distribution.

TLS_GOSTR341094_WITH_28147_CNT_IMIT	GOST94-GOST89-GOST89
TLS_GOSTR341001_WITH_28147_CNT_IMIT	GOST2001-GOST89-GOST89
TLS_GOSTR341094_WITH_NULL_GOSTR3411	GOST94-NULL-GOST94
TLS_GOSTR341001_WITH_NULL_GOSTR3411	GOST2001-NULL-GOST94

Additional Export 1024 and other cipher suites

Note: these ciphers can also be used in SSL v3.

TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA	EXP1024-DES-CBC-SHA
TLS_RSA_EXPORT1024_WITH_RC4_56_SHA	EXP1024-RC4-SHA
TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA	EXP1024-DHE-DSS-DES-CBC-SHA
TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA	EXP1024-DHE-DSS-RC4-SHA
TLS_DHE_DSS_WITH_RC4_128_SHA	DHE-DSS-RC4-SHA

SSL v2.0 cipher suites.

SSL_CK_RC4_128_WITH_MD5	RC4-MD5
SSL_CK_RC4_128_EXPORT40_WITH_MD5	EXP-RC4-MD5
SSL_CK_RC2_128_CBC_WITH_MD5	RC2-MD5
SSL_CK_RC2_128_CBC_EXPORT40_WITH_MD5	EXP-RC2-MD5
SSL_CK_IDEA_128_CBC_WITH_MD5	IDEA-CBC-MD5
SSL_CK_DES_64_CBC_WITH_MD5	DES-CBC-MD5
SSL_CK_DES_192_EDE3_CBC_WITH_MD5	DES-CBC3-MD5

NOTES

The non-ephemeral DH modes are currently unimplemented in OpenSSL because there is no support for DH certificates.

Some compiled versions of OpenSSL may not include all the ciphers listed here because some ciphers were excluded at compile time.

EXAMPLES

Verbose listing of all OpenSSL ciphers including NULL ciphers:

```
 openssl ciphers -v 'ALL:eNULL'
```

Include all ciphers except NULL and anonymous DH then sort by strength:

```
 openssl ciphers -v 'ALL:!ADH:@STRENGTH'
```

Include only 3DES ciphers and then place RSA ciphers last:

```
 openssl ciphers -v '3DES:+RSA'
```

Include all RC4 ciphers but leave out those without authentication:

```
 openssl ciphers -v 'RC4:!COMPLEMENTOFDEFAULT'
```

Include all chiphers with RSA authentication but leave out ciphers without encryption.

```
 openssl ciphers -v 'RSA:!COMPLEMENTOFALL'
```

SEE ALSO

s_client(1), **s_server(1)**, **ssl(3)**

HISTORY

The **COMPLENTOFALL** and **COMPLEMENTOFDEFAULT** selection options for cipherlist strings were added in OpenSSL 0.9.7. The **-V** option for the **ciphers** command was added in OpenSSL 1.0.0.