NAME

SSL_CIPHER_get_cipher_nid, SSL_CIPHER_get_digest_nid, SSL_CIPHER_get_kx_nid, SSL_CIPHER_get_auth_nid, SSL_CIPHER_is_aead, SSL_CIPHER_get_name, SSL_CIPHER_get_bits, SSL_CIPHER_get_version, SSL_CIPHER_description - get SSL_CIPHER properties

SYNOPSIS

```
#include <openssl/ssl.h>

const char *SSL_CIPHER_get_name(const SSL_CIPHER *cipher);
int SSL_CIPHER_get_bits(const SSL_CIPHER *cipher, int *alg_bits);
char *SSL_CIPHER_get_version(const SSL_CIPHER *cipher);
char *SSL_CIPHER_description(const SSL_CIPHER *cipher, char *buf, int size);
int SSL_CIPHER_get_cipher_nid(const SSL_CIPHER *c);
int SSL_CIPHER_get_digest_nid(const SSL_CIPHER *c);
int SSL_CIPHER_get_kx_nid(const SSL_CIPHER *c);
int SSL_CIPHER_get_auth_nid(const SSL_CIPHER *c);
int SSL_CIPHER_is_aead(const SSL_CIPHER *c);
```

DESCRIPTION

SSL_CIPHER_get_name() returns a pointer to the name of **cipher**. If the **cipher** is NULL, it returns ''(NONE)''.

SSL_CIPHER_get_bits() returns the number of secret bits used for **cipher**. If**cipher** is NULL, 0 is returned.

SSL_CIPHER_get_version() returns string which indicates the SSL/TLS protocol version that first defined the cipher. It returns ''(NONE)'' if **cipher** is NULL.

SSL_CIPHER_get_cipher_nid() returns the cipher NID corresponding to **c**. If there is no cipher (e.g. for ciphersuites with no encryption) then **NID_undef** is returned.

SSL_CIPHER_get_digest_nid() returns the digest NID corresponding to the MAC used by **c**. If there is no digest (e.g. for AEAD ciphersuites) then **NID_undef** is returned.

SSL_CIPHER_get_kx_nid() returns the key exchange NID corresponding to the method used by **c**. If there is no key exchange, then **NID_undef** is returned. Examples (not comprehensive):

```
NID_kx_rsa
NID_kx_ecdhe
NID_kx_dhe
NID_kx_psk
```

SSL_CIPHER_get_auth_nid() returns the authentication NID corresponding to the method used by **c**. If there is no authentication, then **NID_undef** is returned. Examples (not comprehensive):

```
NID_auth_rsa
NID_auth_ecdsa
NID_auth_psk
```

SSL_CIPHER_is_aead() returns 1 if the cipher **c** is AEAD (e.g. GCM or ChaCha20/Poly1305), and 0 if it is not AEAD.

SSL_CIPHER_description() returns a textual description of the cipher used into the buffer **buf** of length **len** provided. If **buf** is provided, it must be at least 128 bytes, otherwise a buffer will be allocated using *OPENSSL_malloc()*. If the provided buffer is too small, or the allocation fails, **NULL** is returned.

The string returned by *SSL_CIPHER_description()* consists of several fields separated by whitespace:

<ciphername>
: Textual representation of the cipher name.

<protocol version>
: Protocol version, such as **TLSv1.2**, when the cipher was first defined.

Kx=<key exchange>
: Key exchange method such as **RSA**, **ECDHE**, etc.

Au=<authentication>
: Authentication method such as **RSA**, **None**, etc.. None is the representation of anonymous ciphers.

Enc=<symmetric encryption method>
: Encryption method, with number of secret bits, such as **AESGCM(128)**.

Mac=<message authentication code>
: Message digest, such as **SHA256**.

Some examples for the output of *SSL_CIPHER_description()*:

```
ECDHE-RSA-AES256-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA Enc=AESGCM(256) Mac=AEAD
RSA-PSK-AES256-CBC-SHA384 TLSv1.0 Kx=RSAPSK Au=RSA Enc=AES(256) Mac=SHA384
```

HISTORY

SSL_CIPHER_get_version() was updated to always return the correct protocol string in OpenSSL 1.1.

SSL_CIPHER_description() was changed to return **NULL** on error, rather than a fixed string, in OpenSSL 1.1

SEE ALSO

ssl(3), *SSL_get_current_cipher(3)*, *SSL_get_ciphers(3)*, *ciphers(1)*

COPYRIGHT