NAME

SSL_get1_supported_ciphers, SSL_get_client_ciphers, SSL_get_ciphers, SSL_CTX_get_ciphers, SSL_bytes_to_cipher_list, SSL_get_cipher_list, SSL_get_shared_ciphers - get list of available SSL_CIPHERs

SYNOPSIS

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

STACK_OF(SSL_CIPHER) *SSL_get_ciphers(const SSL *ssl);
STACK_OF(SSL_CIPHER) *SSL_CTX_get_ciphers(const SSL_CTX *ctx);
STACK_OF(SSL_CIPHER) *SSL_get1_supported_ciphers(SSL *s);
STACK_OF(SSL_CIPHER) *SSL_get_client_ciphers(const SSL *ssl);
int SSL_bytes_to_cipher_list(SSL *s, const unsigned char *bytes, size_t len,
                             int isv2format, STACK_OF(SSL_CIPHER) **sk,
                             STACK_OF(SSL_CIPHER) **scsvs);
const char *SSL_get_cipher_list(const SSL *ssl, int priority);
char *SSL_get_shared_ciphers(const SSL *s, char *buf, int size);
```

DESCRIPTION

SSL_get_ciphers() returns the stack of available SSL_CIPHERs for **ssl**, sorted by preference. If **ssl** is NULL or no ciphers are available, NULL is returned.

SSL_CTX_get_ciphers() returns the stack of available SSL_CIPHERs for **ctx**.

SSL_get1_supported_ciphers() returns the stack of enabled SSL_CIPHERs for **ssl** as would be sent in a ClientHello (that is, sorted by preference). The list depends on settings like the cipher list, the supported protocol versions, the security level, and the enabled signature algorithms. SRP and PSK ciphers are only enabled if the appropriate callbacks or settings have been applied. The list of ciphers that would be sent in a ClientHello can differ from the list of ciphers that would be acceptable when acting as a server. For example, additional ciphers may be usable by a server if there is a gap in the list of supported protocols, and some ciphers may not be usable by a server if there is not a suitable certificate configured. If **ssl** is NULL or no ciphers are available, NULL is returned.

SSL_get_client_ciphers() returns the stack of available SSL_CIPHERs matching the list received from the client on **ssl**. If **ssl** is NULL, no ciphers are available, or **ssl** is not operating in server mode, NULL is returned.

SSL_bytes_to_cipher_list() treats the supplied **len** octets in **bytes** as a wire-protocol cipher suite specification (in the three-octet-per-cipher SSLv2 wire format if **isv2format** is nonzero; otherwise the two-octet SSLv3/TLS wire format), and parses the cipher suites supported by the library into the

returned stacks of SSL_CIPHER objects sk and Signalling Cipher-Suite Values scsvs. Unsupported cipher suites are ignored. Returns 1 on success and 0 on failure.

SSL_get_cipher_list() returns a pointer to the name of the SSL_CIPHER listed for **ssl** with **priority**. If **ssl** is NULL, no ciphers are available, or there are less ciphers than **priority** available, NULL is returned.

SSL_get_shared_ciphers() creates a colon separated and NUL terminated list of SSL_CIPHER names that are available in both the client and the server. **buf** is the buffer that should be populated with the list of names and **size** is the size of that buffer. A pointer to **buf** is returned on success or NULL on error. If the supplied buffer is not large enough to contain the complete list of names then a truncated list of names will be returned. Note that just because a ciphersuite is available (i.e. it is configured in the cipher list) and shared by both the client and the server it does not mean that it is enabled (see the description of **SSL_get1_supported_ciphers()** above). This function will return available shared ciphersuites whether or not they are enabled. This is a server side function only and must only be called after the completion of the initial handshake.

NOTES

The details of the ciphers obtained by **SSL_get_ciphers()**, **SSL_CTX_get_ciphers()** **SSL_get1_supported_ciphers()** and **SSL_get_client_ciphers()** can be obtained using the **SSL_CIPHER_get_name**(3) family of functions.

Call **SSL_get_cipher_list()** with **priority** starting from 0 to obtain the sorted list of available ciphers, until NULL is returned.

Note: **SSL_get_ciphers()**, **SSL_CTX_get_ciphers()** and **SSL_get_client_ciphers()** return a pointer to an internal cipher stack, which will be freed later on when the SSL or SSL_SESSION object is freed. Therefore, the calling code **MUST NOT** free the return value itself.

The stack returned by **SSL_get1_supported_ciphers()** should be freed using **sk_SSL_CIPHER_free()**.

The stacks returned by **SSL_bytes_to_cipher_list()** should be freed using **sk_SSL_CIPHER_free()**.

RETURN VALUES

See DESCRIPTION

SEE ALSO

ssl(7), **SSL_CTX_set_cipher_list**(3), **SSL_CIPHER_get_name**(3)

COPYRIGHT