NAME

SSL_CTX_set_cipher_list, SSL_set_cipher_list, SSL_CTX_set_ciphersuites, SSL_set_ciphersuites, OSSL_default_cipher_list, OSSL_default_ciphersuites - choose list of available SSL_CIPHERs

SYNOPSIS

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

int SSL_CTX_set_cipher_list(SSL_CTX *ctx, const char *str);
int SSL_set_cipher_list(SSL *ssl, const char *str);

int SSL_CTX_set_ciphersuites(SSL_CTX *ctx, const char *str);
int SSL_set_ciphersuites(SSL *s, const char *str);

const char *OSSL_default_cipher_list(void);
const char *OSSL_default_ciphersuites(void);
```

DESCRIPTION

SSL_CTX_set_cipher_list() sets the list of available ciphers (TLSv1.2 and below) for **ctx** using the control string **str**. The format of the string is described in **openssl-ciphers**(1). The list of ciphers is inherited by all **ssl** objects created from **ctx**. This function does not impact TLSv1.3 ciphersuites. Use **SSL_CTX_set_ciphersuites()** to configure those.

SSL_set_cipher_list() sets the list of ciphers (TLSv1.2 and below) only for **ssl**.

SSL_CTX_set_ciphersuites() is used to configure the available TLSv1.3 ciphersuites for **ctx**. This is a simple colon (":") separated list of TLSv1.3 ciphersuite names in order of preference. Valid TLSv1.3 ciphersuite names are:

TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256

An empty list is permissible. The default value for the this setting is:

"TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256"

SSL_set_ciphersuites() is the same as **SSL_CTX_set_ciphersuites()** except it configures the ciphersuites for **ssl**.

OSSL_default_cipher_list() returns the default cipher string for TLSv1.2 (and earlier) ciphers.
OSSL_default_ciphersuites() returns the default cipher string for TLSv1.3 ciphersuites.

NOTES

The control string **str** for **SSL_CTX_set_cipher_list()**, **SSL_set_cipher_list()**, **SSL_CTX_set_ciphersuites()** and **SSL_set_ciphersuites()** should be universally usable and not depend on details of the library configuration (ciphers compiled in). Thus no syntax checking takes place. Items that are not recognized, because the corresponding ciphers are not compiled in or because they are mistyped, are simply ignored. Failure is only flagged if no ciphers could be collected at all.

It should be noted, that inclusion of a cipher to be used into the list is a necessary condition. On the client side, the inclusion into the list is also sufficient unless the security level excludes it. On the server side, additional restrictions apply. All ciphers have additional requirements. ADH ciphers don't need a certificate, but DH-parameters must have been set. All other ciphers need a corresponding certificate and key.

An RSA cipher can only be chosen, when an RSA certificate is available. RSA ciphers using DHE need a certificate and key and additional DH-parameters (see **SSL_CTX_set_tmp_dh_callback**(3)).

A DSA cipher can only be chosen, when a DSA certificate is available. DSA ciphers always use DH key exchange and therefore need DH-parameters (see **SSL_CTX_set_tmp_dh_callback**(3)).

When these conditions are not met for any cipher in the list (e.g. a client only supports export RSA ciphers with an asymmetric key length of 512 bits and the server is not configured to use temporary RSA keys), the "no shared cipher" (SSL_R_NO_SHARED_CIPHER) error is generated and the handshake will fail.

OSSL_default_cipher_list() and **OSSL_default_ciphersuites()** replace SSL_DEFAULT_CIPHER_LIST and TLS_DEFAULT_CIPHERSUITES, respectively. The cipher list defines are deprecated as of 3.0.

RETURN VALUES

SSL_CTX_set_cipher_list() and **SSL_set_cipher_list()** return 1 if any cipher could be selected and 0 on complete failure.

SSL_CTX_set_ciphersuites() and **SSL_set_ciphersuites()** return 1 if the requested ciphersuite list was configured, and 0 otherwise.

SEE ALSO

ssl(7), **SSL_get_ciphers**(3), **SSL_CTX_use_certificate**(3), **SSL_CTX_set_tmp_dh_callback**(3),

openssl-ciphers(1)

HISTORY

OSSL_default_cipher_list() and **OSSL_default_ciphersites()** are new in 3.0.

COPYRIGHT