NAME
>   CURLOPT_SSL_CTX_FUNCTION – openssl specific callback to do SSL magic

SYNOPSIS
>   #include <curl/curl.h>
>
>   CURLcode ssl_ctx_callback(CURL *curl, void *ssl_ctx, void *userptr);
>
>   CURLcode curl_easy_setopt(CURL *handle, CURLOPT_SSL_CTX_FUNCTION,
>   ssl_ctx_callback);

DESCRIPTION
>   This option only works for libcurl powered by OpenSSL. If libcurl was built against another SSL library, this functionality is absent.
>
>   Pass a pointer to your callback function, which should match the prototype shown above.
>
>   This callback function gets called by libcurl just before the initialization of a SSL connection after having processed all other SSL related options to give a last chance to an application to modify the behaviour of openssl's ssl initialization. The *sslctx* parameter is actually a pointer to an openssl *SSL_CTX*. If an error is returned from the callback, no attempt to establish a connection is made and the perform operation will return the error code. Set the *userptr* argument with the *CURLOPT_SSL_CTX_DATA(3)* option.
>
>   This function will get called on all new connections made to a server, during the SSL negotiation. The SSL_CTX pointer will be a new one every time.
>
>   To use this properly, a non-trivial amount of knowledge of the openssl libraries is necessary. For example, using this function allows you to use openssl callbacks to add additional validation code for certificates, and even to change the actual URI of a HTTPS request (example used in the lib509 test case). See also the example section for a replacement of the key, certificate and trust file settings.

DEFAULT
>   NULL

PROTOCOLS
>   All TLS based protocols: HTTPS, FTPS, IMAPS, POP3, SMTPS etc.

EXAMPLE
>   TODO

AVAILABILITY
>   Added in 7.11.0. Only supported when built with OpenSSL.

RETURN VALUE
>   Returns CURLE_OK if the option is supported, and CURLE_UNKNOWN_OPTION if not.

SEE ALSO
>   **CURLOPT_SSL_CTX_DATA**(3), **CURLOPT_SSL_VERIFYPEER**(3),