NAME

SSL_export_keying_material, SSL_export_keying_material_early - obtain keying material for application use

SYNOPSIS

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

int SSL_export_keying_material(SSL *s, unsigned char *out, size_t olen,
                               const char *label, size_t llen,
                               const unsigned char *context,
                               size_t contextlen, int use_context);

int SSL_export_keying_material_early(SSL *s, unsigned char *out, size_t olen,
                                     const char *label, size_t llen,
                                     const unsigned char *context,
                                     size_t contextlen);
```

DESCRIPTION

During the creation of a TLS or DTLS connection shared keying material is established between the two endpoints. The functions **SSL_export_keying_material()** and **SSL_export_keying_material_early()** enable an application to use some of this keying material for its own purposes in accordance with RFC5705 (for TLSv1.2 and below) or RFC8446 (for TLSv1.3).

SSL_export_keying_material() derives keying material using the *exporter_master_secret* established in the handshake.

SSL_export_keying_material_early() is only usable with TLSv1.3, and derives keying material using the *early_exporter_master_secret* (as defined in the TLS 1.3 RFC). For the client, the *early_exporter_master_secret* is only available when the client attempts to send 0-RTT data. For the server, it is only available when the server accepts 0-RTT data.

An application may need to securely establish the context within which this keying material will be used. For example this may include identifiers for the application session, application algorithms or parameters, or the lifetime of the context. The context value is left to the application but must be the same on both sides of the communication.

For a given SSL connection **s**, **olen** bytes of data will be written to **out**. The application specific context should be supplied in the location pointed to by **context** and should be **contextlen** bytes long. Provision of a context is optional. If the context should be omitted entirely then **use_context** should be set to 0. Otherwise it should be any other value. If **use_context** is 0 then the values of **context** and **contextlen** are

ignored. Note that in TLSv1.2 and below a zero length context is treated differently from no context at all, and will result in different keying material being returned. In TLSv1.3 a zero length context is that same as no context at all and will result in the same keying material being returned.

An application specific label should be provided in the location pointed to by **label** and should be **llen** bytes long. Typically this will be a value from the IANA Exporter Label Registry (<https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#exporter-labels>). Alternatively labels beginning with "EXPERIMENTAL" are permitted by the standard to be used without registration. TLSv1.3 imposes a maximum label length of 249 bytes.

Note that this function is only defined for TLSv1.0 and above, and DTLSv1.0 and above. Attempting to use it in SSLv3 will result in an error.

RETURN VALUES

SSL_export_keying_material() returns 0 or -1 on failure or 1 on success.

SSL_export_keying_material_early() returns 0 on failure or 1 on success.

SEE ALSO

ssl(7)

HISTORY

The **SSL_export_keying_material_early()** function was added in OpenSSL 1.1.1.

COPYRIGHT