NAME

EVP_KDF−TLS13_KDF − The TLS 1.3 EVP_KDF implementation

DESCRIPTION

Support for computing the TLS 1.3 version of the **HKDF** KDF through the **EVP_KDF** API.

The EVP_KDF−TLS13_KDF algorithm implements the HKDF key derivation function as used by TLS 1.3.

Identity

‘‘TLS13−KDF’’ is the name for this implementation; it can be used with the **EVP_KDF_fetch()** function.

Supported parameters

The supported parameters are:

‘‘properties’’ (**OSSL_KDF_PARAM_PROPERTIES**) <UTF8 string>
‘‘digest’’ (**OSSL_KDF_PARAM_DIGEST**) <UTF8 string>
‘‘key’’ (**OSSL_KDF_PARAM_KEY**) <octet string>
‘‘salt’’ (**OSSL_KDF_PARAM_SALT**) <octet string>

These parameters work as described in ‘‘PARAMETERS’’ in **EVP_KDF**(3).

‘‘prefix’’ (**OSSL_KDF_PARAM_PREFIX**) <octet string>

This parameter sets the label prefix on the specified TLS 1.3 KDF context. For TLS 1.3 this should be set to the ASCII string ‘‘tls13 ’’ without a trailing zero byte. Refer to RFC 8446 section 7.1 ‘‘Key Schedule’’ for details.

‘‘label’’ (**OSSL_KDF_PARAM_LABEL**) <octet string>

This parameter sets the label on the specified TLS 1.3 KDF context. Refer to RFC 8446 section 7.1 ‘‘Key Schedule’’ for details.

‘‘data’’ (**OSSL_KDF_PARAM_DATA**) <octet string>

This parameter sets the context data on the specified TLS 1.3 KDF context. Refer to RFC 8446 section 7.1 ‘‘Key Schedule’’ for details.

‘‘mode’’ (**OSSL_KDF_PARAM_MODE**) <UTF8 string> or <integer>

This parameter sets the mode for the TLS 1.3 KDF operation. There are two modes that are currently defined:

‘‘EXTRACT_ONLY’’ or **EVP_KDF_HKDF_MODE_EXTRACT_ONLY**

In this mode calling **EVP_KDF_derive**(3) will just perform the extract operation. The value returned will be the intermediate fixed-length pseudorandom key K. The *keylen* parameter must match the size of K, which can be looked up by calling **EVP_KDF_CTX_get_kdf_size()** after setting the mode and digest.

The digest, key and salt values must be set before a key is derived otherwise an error will occur.

‘‘EXPAND_ONLY’’ or **EVP_KDF_HKDF_MODE_EXPAND_ONLY**

In this mode calling **EVP_KDF_derive**(3) will just perform the expand operation. The input key should be set to the intermediate fixed-length pseudorandom key K returned from a previous extract operation.

The digest, key and info values must be set before a key is derived otherwise an error will occur.

NOTES

This KDF is intended for use by the TLS 1.3 implementation in libssl. It does not support all the options and capabilities that HKDF does.

The *OSSL_PARAM* array passed to **EVP_KDF_derive**(3) or **EVP_KDF_CTX_set_params**(3) must specify all of the parameters required. This KDF does not support a piecemeal approach to providing these.

A context for a TLS 1.3 KDF can be obtained by calling:

```
EVP_KDF *kdf = EVP_KDF_fetch(NULL, "TLS13−KDF", NULL);
EVP_KDF_CTX *kctx = EVP_KDF_CTX_new(kdf);
```

The output length of a TLS 1.3 KDF expand operation is specified via the *keylen* parameter to the

EVP_KDF_derive(3) function. When using EVP_KDF_HKDF_MODE_EXTRACT_ONLY the *keylen* parameter must equal the size of the intermediate fixed-length pseudorandom key otherwise an error will occur. For that mode, the fixed output size can be looked up by calling **EVP_KDF_CTX_get_kdf_size()** after setting the mode and digest on the **EVP_KDF_CTX**.

CONFORMING TO

RFC 8446

SEE ALSO

EVP_KDF(3), **EVP_KDF_CTX_new**(3), **EVP_KDF_CTX_free**(3), **EVP_KDF_CTX_get_kdf_size**(3), **EVP_KDF_CTX_set_params**(3), **EVP_KDF_derive**(3), ''PARAMETERS'' in **EVP_KDF**(3), **EVP_KDF−HKDF**(7)

COPYRIGHT