Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'EVP_KDF-TLS1_PRF.7ossl' command

$ man EVP_KDF-TLS1_PRF.7ossl

EVP_KDF-TLS1_PRF(7ossl) OpenSSL EVP_KDF-TLS1_PRF(7ossl)

NAME

EVP_KDF-TLS1_PRF - The TLS1 PRF EVP_KDF implementation

DESCRIPTION

Support for computing the TLS1 PRF through the EVP_KDF API.

The EVP_KDF-TLS1_PRF algorithm implements the PRF used by TLS versions up to and including TLS 1.2.

Identity

"TLS1-PRF" 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>

These parameters work as described in "PARAMETERS" in EVP_KDF(3).

The OSSL_KDF_PARAM_DIGEST parameter is used to set the message

digest associated with the TLS PRF. EVP_md5_sha1() is treated as a special case which uses the PRF algorithm using both MD5 and SHA1 as used in TLS 1.0 and 1.1.

"secret" (OSSL_KDF_PARAM_SECRET) <octet string>

This parameter sets the secret value of the TLS PRF. Any existing secret value is replaced.

"seed" (OSSL_KDF_PARAM_SEED) <octet string>

This parameter sets the context seed. The length of the context seed cannot exceed 1024 bytes; this should be more than enough for any normal use of the TLS PRF.

NOTES

A context for the TLS PRF can be obtained by calling:

```
EVP_KDF *kdf = EVP_KDF_fetch(NULL, "TLS1-PRF", NULL);
EVP_KDF_CTX *kctx = EVP_KDF_CTX_new(kdf);
```

The digest, secret value and seed must be set before a key is derived otherwise an error will occur.

The output length of the PRF is specified by the keylen parameter to the EVP_KDF_derive() function.

EXAMPLES

This example derives 10 bytes using SHA-256 with the secret key "secret" and seed value "seed":

```
EVP_KDF *kdf;
EVP_KDF_CTX *kctx;
unsigned char out[10];
OSSL_PARAM params[4], *p = params;
```

```
kdf = EVP_KDF_fetch(NULL, "TLS1-PRF", NULL);
kctx = EVP_KDF_CTX_new(kdf);
EVP_KDF_free(kdf);

*p++ = OSSL_PARAM_construct_utf8_string(OSSL_KDF_PARAM_DIGEST,
                                        SN_sha256, strlen(SN_sha256));
*p++ = OSSL_PARAM_construct_octet_string(OSSL_KDF_PARAM_SECRET,
                                         "secret", (size_t)6);
*p++ = OSSL_PARAM_construct_octet_string(OSSL_KDF_PARAM_SEED,
                                         "seed", (size_t)4);
*p = OSSL_PARAM_construct_end();
if (EVP_KDF_derive(kctx, out, sizeof(out), params) <= 0) {
    error("EVP_KDF_derive");
}
EVP_KDF_CTX_free(kctx);
```

CONFORMING TO

RFC 2246, RFC 5246 and NIST SP 800-135 r1

SEE ALSO

EVP_KDF(3), EVP_KDF_CTX_new(3), EVP_KDF_CTX_free(3),
EVP_KDF_CTX_set_params(3), EVP_KDF_derive(3), "PARAMETERS" in
EVP_KDF(3)

COPYRIGHT