NAME

EVP_PKEY_CTX_set_tls1_prf_md, EVP_PKEY_CTX_set1_tls1_prf_secret, EVP_PKEY_CTX_add1_tls1_prf_seed - TLS PRF key derivation algorithm

SYNOPSIS

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

int EVP_PKEY_CTX_set_tls1_prf_md(EVP_PKEY_CTX *pctx, const EVP_MD *md);
int EVP_PKEY_CTX_set1_tls1_prf_secret(EVP_PKEY_CTX *pctx,
                                      unsigned char *sec, int seclen);
int EVP_PKEY_CTX_add1_tls1_prf_seed(EVP_PKEY_CTX *pctx,
                                    unsigned char *seed, int seedlen);
```

DESCRIPTION

The **EVP_PKEY_TLS1_PRF** algorithm implements the PRF key derivation function for TLS. It has no associated private key and only implements key derivation using **EVP_PKEY_derive**(3).

EVP_PKEY_set_tls1_prf_md() sets 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.

EVP_PKEY_CTX_set_tls1_prf_secret() sets the secret value of the TLS PRF to **seclen** bytes of the buffer **sec**. Any existing secret value is replaced and any seed is reset.

EVP_PKEY_CTX_add1_tls1_prf_seed() sets the seed to **seedlen** bytes of **seed**. If a seed is already set it is appended to the existing value.

STRING CTRLS

The TLS PRF also supports string based control operations using **EVP_PKEY_CTX_ctrl_str**(3). The **type** parameter "md" uses the supplied **value** as the name of the digest algorithm to use. The **type** parameters "secret" and "seed" use the supplied **value** parameter as a secret or seed value. The names "hexsecret" and "hexseed" are similar except they take a hex string which is converted to binary.

NOTES

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

```
EVP_PKEY_CTX *pctx = EVP_PKEY_CTX_new_id(EVP_PKEY_TLS1_PRF, NULL);
```

The digest, secret value and seed must be set before a key is derived or an error occurs.

The total length of all seeds cannot exceed 1024 bytes in length: this should be more than enough for any normal use of the TLS PRF.

The output length of the PRF is specified by the length parameter in the **EVP_PKEY_derive()** function. Since the output length is variable, setting the buffer to **NULL** is not meaningful for the TLS PRF.

Optimised versions of the TLS PRF can be implemented in an ENGINE.

RETURN VALUES

All these functions return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

EXAMPLES

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

```
 EVP_PKEY_CTX *pctx;
 unsigned char out[10];
 size_t outlen = sizeof(out);

 pctx = EVP_PKEY_CTX_new_id(EVP_PKEY_TLS1_PRF, NULL);
 if (EVP_PKEY_derive_init(pctx) <= 0)
     /* Error */
 if (EVP_PKEY_CTX_set_tls1_prf_md(pctx, EVP_sha256()) <= 0)
     /* Error */
 if (EVP_PKEY_CTX_set1_tls1_prf_secret(pctx, "secret", 6) <= 0)
     /* Error */
 if (EVP_PKEY_CTX_add1_tls1_prf_seed(pctx, "seed", 4) <= 0)
     /* Error */
 if (EVP_PKEY_derive(pctx, out, &outlen) <= 0)
     /* Error */
```

SEE ALSO

EVP_PKEY_CTX_new(3), **EVP_PKEY_CTX_ctrl_str**(3), **EVP_PKEY_derive**(3)

HISTORY

All of the functions described here were converted from macros to functions in OpenSSL 3.0.

COPYRIGHT