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 'provider-keyexch.7ossl' command

$ man provider-keyexch.7ossl

PROVIDER-KEYEXCH(7ossl) OpenSSL PROVIDER-KEYEXCH(7ossl)

NAME

provider-keyexch - The keyexch library <-> provider functions

SYNOPSIS

```
#include <openssl/core_dispatch.h>
#include <openssl/core_names.h>
/*
 * None of these are actual functions, but are displayed like this for
 * the function signatures for functions that are offered as function
 * pointers in OSSL_DISPATCH arrays.
 */
/* Context management */
void *OSSL_FUNC_keyexch_newctx(void *provctx);
void OSSL_FUNC_keyexch_freectx(void *ctx);
void *OSSL_FUNC_keyexch_dupctx(void *ctx);
/* Shared secret derivation */
int OSSL_FUNC_keyexch_init(void *ctx, void *provkey,
                           const OSSL_PARAM params[]);
int OSSL_FUNC_keyexch_set_peer(void *ctx, void *provkey);
int OSSL_FUNC_keyexch_derive(void *ctx, unsigned char *secret, size_t *secretlen,
                             size_t outlen);
/* Key Exchange parameters */
int OSSL_FUNC_keyexch_set_ctx_params(void *ctx, const OSSL_PARAM params[]);
```

```
const OSSL_PARAM *OSSL_FUNC_keyexch_settable_ctx_params(void *ctx,
                                                        void *provctx);
int OSSL_FUNC_keyexch_get_ctx_params(void *ctx, OSSL_PARAM params[]);
const OSSL_PARAM *OSSL_FUNC_keyexch_gettable_ctx_params(void *ctx,
                                                        void *provctx);
```

DESCRIPTION

This documentation is primarily aimed at provider authors. See provider(7) for further information.

The key exchange (OSSL_OP_KEYEXCH) operation enables providers to implement key exchange algorithms and make them available to applications via EVP_PKEY_derive(3) and other related functions).

All "functions" mentioned here are passed as function pointers between libcrypto and the provider in OSSL_DISPATCH arrays via OSSL_ALGORITHM arrays that are returned by the provider's provider_query_operation() function (see "Provider Functions" in provider-base(7)).

All these "functions" have a corresponding function type definition named OSSL_FUNC_{name}_fn, and a helper function to retrieve the function pointer from an OSSL_DISPATCH element named OSSL_FUNC_{name}.

For example, the "function" OSSL_FUNC_keyexch_newctx() has these:

```
typedef void *(OSSL_FUNC_keyexch_newctx_fn)(void *provctx);
static ossl_inline OSSL_FUNC_keyexch_newctx_fn
    OSSL_FUNC_keyexch_newctx(const OSSL_DISPATCH *opf);
```

OSSL_DISPATCH arrays are indexed by numbers that are provided as macros in openssl-core_dispatch.h(7), as follows:

```
OSSL_FUNC_keyexch_newctx              OSSL_FUNC_KEYEXCH_NEWCTX
OSSL_FUNC_keyexch_freectx             OSSL_FUNC_KEYEXCH_FREECTX
OSSL_FUNC_keyexch_dupctx              OSSL_FUNC_KEYEXCH_DUPCTX
OSSL_FUNC_keyexch_init                OSSL_FUNC_KEYEXCH_INIT
OSSL_FUNC_keyexch_set_peer            OSSL_FUNC_KEYEXCH_SET_PEER
OSSL_FUNC_keyexch_derive              OSSL_FUNC_KEYEXCH_DERIVE
OSSL_FUNC_keyexch_set_ctx_params      OSSL_FUNC_KEYEXCH_SET_CTX_PARAMS
OSSL_FUNC_keyexch_settable_ctx_params OSSL_FUNC_KEYEXCH_SETTABLE_CTX_PARAMS
OSSL_FUNC_keyexch_get_ctx_params      OSSL_FUNC_KEYEXCH_GET_CTX_PARAMS
```

OSSL_FUNC_keyexch_gettable_ctx_params OSSL_FUNC_KEYEXCH_GETTABLE_CTX_PARAMS

A key exchange algorithm implementation may not implement all of these functions. In order to be a consistent set of functions a provider must implement OSSL_FUNC_keyexch_newctx, OSSL_FUNC_keyexch_freectx, OSSL_FUNC_keyexch_init and OSSL_FUNC_keyexch_derive. All other functions are optional.

A key exchange algorithm must also implement some mechanism for generating, loading or importing keys via the key management (OSSL_OP_KEYMGMT) operation. See provider-keymgmt(7) for further details.

Context Management Functions

OSSL_FUNC_keyexch_newctx() should create and return a pointer to a provider side structure for holding context information during a key exchange operation. A pointer to this context will be passed back in a number of the other key exchange operation function calls. The parameter provctx is the provider context generated during provider initialisation (see provider(7)).

OSSL_FUNC_keyexch_freectx() is passed a pointer to the provider side key exchange context in the ctx parameter. This function should free any resources associated with that context.

OSSL_FUNC_keyexch_dupctx() should duplicate the provider side key exchange context in the ctx parameter and return the duplicate copy.

Shared Secret Derivation Functions

OSSL_FUNC_keyexch_init() initialises a key exchange operation given a provider side key exchange context in the ctx parameter, and a pointer to a provider key object in the provkey parameter. The params, if not NULL, should be set on the context in a manner similar to using OSSL_FUNC_keyexch_set_params(). The key object should have been previously generated, loaded or imported into the provider using the key management (OSSL_OP_KEYMGMT) operation (see provider-keymgmt(7)>.

OSSL_FUNC_keyexch_set_peer() is called to supply the peer's public key (in the provkey parameter) to be used when deriving the shared secret. It is also passed a previously initialised key exchange context in the

ctx parameter. The key object should have been previously generated, loaded or imported into the provider using the key management (OSSL_OP_KEYMGMT) operation (see provider-keymgmt(7)>.

OSSL_FUNC_keyexch_derive() performs the actual key exchange itself by deriving a shared secret. A previously initialised key exchange context is passed in the ctx parameter. The derived secret should be written to the location secret which should not exceed outlen bytes. The length of the shared secret should be written to *secretlen. If secret is NULL then the maximum length of the shared secret should be written to *secretlen.

Key Exchange Parameters Functions

OSSL_FUNC_keyexch_set_ctx_params() sets key exchange parameters associated with the given provider side key exchange context ctx to params, see "Common Key Exchange parameters". Any parameter settings are additional to any that were previously set. Passing NULL for params should return true.

OSSL_FUNC_keyexch_get_ctx_params() gets key exchange parameters associated with the given provider side key exchange context ctx into params, see "Common Key Exchange parameters". Passing NULL for params should return true.

OSSL_FUNC_keyexch_settable_ctx_params() yields a constant OSSL_PARAM array that describes the settable parameters, i.e. parameters that can be used with OP_signature_set_ctx_params(). If OSSL_FUNC_keyexch_settable_ctx_params() is present, OSSL_FUNC_keyexch_set_ctx_params() must also be present, and vice versa. Similarly, OSSL_FUNC_keyexch_gettable_ctx_params() yields a constant OSSL_PARAM array that describes the gettable parameters, i.e. parameters that can be handled by OP_signature_get_ctx_params(). If OSSL_FUNC_keyexch_gettable_ctx_params() is present, OSSL_FUNC_keyexch_get_ctx_params() must also be present, and vice versa. See OSSL_PARAM(3) for the use of OSSL_PARAM as parameter descriptor.

Notice that not all settable parameters are also gettable, and vice

versa.

Common Key Exchange parameters

See OSSL_PARAM(3) for further details on the parameters structure used by the OSSL_FUNC_keyexch_set_ctx_params() and OSSL_FUNC_keyexch_get_ctx_params() functions.

Common parameters currently recognised by built-in key exchange algorithms are as follows.

"kdf-type" (OSSL_EXCHANGE_PARAM_KDF_TYPE) <UTF8 string>

Sets or gets the Key Derivation Function type to apply within the associated key exchange ctx.

"kdf-digest" (OSSL_EXCHANGE_PARAM_KDF_DIGEST) <UTF8 string>

Sets or gets the Digest algorithm to be used as part of the Key Derivation Function associated with the given key exchange ctx.

"kdf-digest-props" (OSSL_EXCHANGE_PARAM_KDF_DIGEST_PROPS) <UTF8 string>

Sets properties to be used upon look up of the implementation for the selected Digest algorithm for the Key Derivation Function associated with the given key exchange ctx.

"kdf-outlen" (OSSL_EXCHANGE_PARAM_KDF_OUTLEN) <unsigned integer>

Sets or gets the desired size for the output of the chosen Key Derivation Function associated with the given key exchange ctx. The length of the "kdf-outlen" parameter should not exceed that of a size_t.

"kdf-ukm" (OSSL_EXCHANGE_PARAM_KDF_UKM) <octet string>

Sets the User Key Material to be used as part of the selected Key Derivation Function associated with the given key exchange ctx.

"kdf-ukm" (OSSL_EXCHANGE_PARAM_KDF_UKM) <octet string ptr>

Gets a pointer to the User Key Material to be used as part of the selected Key Derivation Function associated with the given key exchange ctx. Providers usually do not need to support this gettable parameter as its sole purpose is to support functionality of the deprecated EVP_PKEY_CTX_get0_ecdh_kdf_ukm() and EVP_PKEY_CTX_get0_dh_kdf_ukm() functions.

RETURN VALUES

OSSL_FUNC_keyexch_newctx() and OSSL_FUNC_keyexch_dupctx() should return the newly created provider side key exchange context, or NULL on failure.

OSSL_FUNC_keyexch_init(), OSSL_FUNC_keyexch_set_peer(), OSSL_FUNC_keyexch_derive(), OSSL_FUNC_keyexch_set_params(), and OSSL_FUNC_keyexch_get_params() should return 1 for success or 0 on error.

OSSL_FUNC_keyexch_settable_ctx_params() and OSSL_FUNC_keyexch_gettable_ctx_params() should always return a constant OSSL_PARAM array.

SEE ALSO

provider(7)

HISTORY

The provider KEYEXCH interface was introduced in OpenSSL 3.0.

COPYRIGHT