



**Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!**

***Rocky Enterprise Linux 9.2 Manual Pages on command 'provider-keyexch.7ssl'***

***\$ man provider-keyexch.7ssl***

PROVIDER-KEYEXCH(7SSL)                      OpenSSL                      PROVIDER-KEYEXCH(7SSL)

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

Copyright 2019-2022 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <<https://www.openssl.org/source/license.html>>.