



**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 'EVP\_PKEY-X448.7ssl'***

***\$ man EVP\_PKEY-X448.7ssl***

EVP\_PKEY-X25519(7SSL)                      OpenSSL                      EVP\_PKEY-X25519(7SSL)

**NAME**

EVP\_PKEY-X25519, EVP\_PKEY-X448, EVP\_PKEY-ED25519, EVP\_PKEY-ED448, EVP\_KEYMGMT-X25519, EVP\_KEYMGMT-X448, EVP\_KEYMGMT-ED25519, EVP\_KEYMGMT-ED448 - EVP\_PKEY X25519, X448, ED25519 and ED448 keytype and algorithm support

**DESCRIPTION**

The X25519, X448, ED25519 and ED448 keytypes are implemented in OpenSSL's default and FIPS providers. These implementations support the associated key, containing the public key pub and the private key priv.

No additional parameters can be set during key generation.

**Common X25519, X448, ED25519 and ED448 parameters**

In addition to the common parameters that all keytypes should support (see "Common parameters" in provider-keymgmt(7)), the implementation of these keytypes support the following.

"group" (OSSL\_PKEY\_PARAM\_GROUP\_NAME) <UTF8 string>

This is only supported by X25519 and X448. The group name must be "x25519" or "x448" respectively for those algorithms. This is only present for consistency with other key exchange algorithms and is typically not needed.

"pub" (OSSL\_PKEY\_PARAM\_PUB\_KEY) <octet string>

The public key value.

"priv" (OSSL\_PKEY\_PARAM\_PRIV\_KEY) <octet string>

The private key value.

"encoded-pub-key" (OSSL\_PKEY\_PARAM\_ENCODED\_PUBLIC\_KEY) <octet string>

Used for getting and setting the encoding of a public key for the X25519 and X448 key types. Public keys are expected be encoded in a format as defined by RFC7748.

ED25519 and ED448 parameters

"mandatory-digest" (OSSL\_PKEY\_PARAM\_MANDATORY\_DIGEST) <UTF8 string>

The empty string, signifying that no digest may be specified.

## CONFORMING TO

RFC 8032

RFC 8410

## EXAMPLES

An EVP\_PKEY context can be obtained by calling:

```
EVP_PKEY_CTX *pctx =
    EVP_PKEY_CTX_new_from_name(NULL, "X25519", NULL);
EVP_PKEY_CTX *pctx =
    EVP_PKEY_CTX_new_from_name(NULL, "X448", NULL);
EVP_PKEY_CTX *pctx =
    EVP_PKEY_CTX_new_from_name(NULL, "ED25519", NULL);
EVP_PKEY_CTX *pctx =
    EVP_PKEY_CTX_new_from_name(NULL, "ED448", NULL);
```

An X25519 key can be generated like this:

```
pkey = EVP_PKEY_Q_keygen(NULL, NULL, "X25519");
```

An X448, ED25519, or ED448 key can be generated likewise.

## SEE ALSO

EVP\_KEYMGMT(3), EVP\_PKEY(3), provider-keymgmt(7), EVP\_KEYEXCH-X25519(7),  
EVP\_KEYEXCH-X448(7), EVP\_SIGNATURE-ED25519(7), EVP\_SIGNATURE-ED448(7)

## COPYRIGHT

Copyright 2020-2021 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>>.