NAME

openssl-kdf - perform Key Derivation Function operations

SYNOPSIS

openssl kdf [**-help**] [**-cipher**] [**-digest**] [**-mac**] [**-kdfopt** *nm*:*v*] [**-keylen** *num*] [**-out** *filename*] [**-binary**] [**-provider** *name*] [**-provider-path** *path*] [**-propquery** *propq*] *kdf_name*

DESCRIPTION

The key derivation functions generate a derived key from either a secret or password.

OPTIONS

-help

Print a usage message.

-keylen *num*

The output size of the derived key. This field is required.

-out *filename*

Filename to output to, or standard output by default.

-binary

Output the derived key in binary form. Uses hexadecimal text format if not specified.

-cipher *name*

Specify the cipher to be used by the KDF. Not all KDFs require a cipher and it is an error to use this option in such cases.

-digest *name*

Specify the digest to be used by the KDF. Not all KDFs require a digest and it is an error to use this option in such cases. To see the list of supported digests, use "openssl list -digest-commands".

-mac *name*

Specify the MAC to be used by the KDF. Not all KDFs require a MAC and it is an error to use this option in such cases.

-kdfopt *nm*:*v*

Passes options to the KDF algorithm. A comprehensive list of parameters can be found in "PARAMETERS" in **EVP_KDF**(3). Common parameter names used by **EVP_KDF_CTX_set_params()** are:

key:*string*

Specifies the secret key as an alphanumeric string (use if the key contains printable characters only). The string length must conform to any restrictions of the KDF algorithm. A key must be specified for most KDF algorithms.

hexkey:*string*

Alternative to the **key:** option where the secret key is specified in hexadecimal form (two hex digits per byte).

pass:*string*

Specifies the password as an alphanumeric string (use if the password contains printable characters only). The password must be specified for PBKDF2 and scrypt.

hexpass:*string*

Alternative to the **pass:** option where the password is specified in hexadecimal form (two hex digits per byte).

salt:*string*

Specifies a non-secret unique cryptographic salt as an alphanumeric string (use if it contains printable characters only). The length must conform to any restrictions of the KDF algorithm. A salt parameter is required for several KDF algorithms, such as **EVP_KDF-PBKDF2**(7).

hexsalt:*string*

Alternative to the **salt:** option where the salt is specified in hexadecimal form (two hex digits per byte).

info:*string*

Some KDF implementations, such as **EVP_KDF-HKDF**(7), take an 'info' parameter for binding the derived key material to application- and context-specific information. Specifies the info, fixed info, other info or shared info argument as an alphanumeric string (use if it contains printable characters only). The length must conform to any restrictions of the KDF algorithm.

hexinfo:*string*

Alternative to the **info:** option where the info is specified in hexadecimal form (two hex digits per byte).

digest:*string*

This option is identical to the **-digest** option.

cipher:*string*
This option is identical to the **-cipher** option.

mac:*string*
This option is identical to the **-mac** option.

-provider *name*
-provider-path *path*
-propquery *propq*
See "Provider Options" in **openssl**(1), **provider**(7), and **property**(7).

kdf_name
Specifies the name of a supported KDF algorithm which will be used. The supported algorithms names include TLS1-PRF, HKDF, SSKDF, PBKDF2, SSHKDF, X942KDF-ASN1, X942KDF-CONCAT, X963KDF and SCRYPT.

EXAMPLES

Use TLS1-PRF to create a hex-encoded derived key from a secret key and seed:

```
openssl kdf -keylen 16 -kdfopt digest:SHA2-256 -kdfopt key:secret \
    -kdfopt seed:seed TLS1-PRF
```

Use HKDF to create a hex-encoded derived key from a secret key, salt and info:

```
openssl kdf -keylen 10 -kdfopt digest:SHA2-256 -kdfopt key:secret \
    -kdfopt salt:salt -kdfopt info:label HKDF
```

Use SSKDF with KMAC to create a hex-encoded derived key from a secret key, salt and info:

```
openssl kdf -keylen 64 -kdfopt mac:KMAC-128 -kdfopt maclen:20 \
    -kdfopt hexkey:b74a149a161545 -kdfopt hexinfo:348a37a2 \
    -kdfopt hexsalt:3638271ccd68a2 SSKDF
```

Use SSKDF with HMAC to create a hex-encoded derived key from a secret key, salt and info:

```
openssl kdf -keylen 16 -kdfopt mac:HMAC -kdfopt digest:SHA2-256 \
    -kdfopt hexkey:b74a149a -kdfopt hexinfo:348a37a2 \
    -kdfopt hexsalt:3638271c SSKDF
```

Use SSKDF with Hash to create a hex-encoded derived key from a secret key, salt and info:

```
openssl kdf -keylen 14 -kdfopt digest:SHA2-256 \
        -kdfopt hexkey:6dbdc23f045488 \
        -kdfopt hexinfo:a1b2c3d4 SSKDF
```

Use SSHKDF to create a hex-encoded derived key from a secret key, hash and session_id:

```
openssl kdf -keylen 16 -kdfopt digest:SHA2-256 \
        -kdfopt hexkey:0102030405 \
        -kdfopt hexxcghash:06090A \
        -kdfopt hexsession_id:01020304 \
        -kdfopt type:A SSHKDF
```

Use PBKDF2 to create a hex-encoded derived key from a password and salt:

```
openssl kdf -keylen 32 -kdfopt digest:SHA256 -kdfopt pass:password \
        -kdfopt salt:salt -kdfopt iter:2 PBKDF2
```

Use scrypt to create a hex-encoded derived key from a password and salt:

```
openssl kdf -keylen 64 -kdfopt pass:password -kdfopt salt:NaCl \
        -kdfopt n:1024 -kdfopt r:8 -kdfopt p:16 \
        -kdfopt maxmem_bytes:10485760 SCRYPT
```

NOTES

The KDF mechanisms that are available will depend on the options used when building OpenSSL.

SEE ALSO

openssl(1), **openssl-pkeyutl**(1), **EVP_KDF**(3), **EVP_KDF-SCRYPT**(7), **EVP_KDF-TLS1_PRF**(7), **EVP_KDF-PBKDF2**(7), **EVP_KDF-HKDF**(7), **EVP_KDF-SS**(7), **EVP_KDF-SSHKDF**(7), **EVP_KDF-X942-ASN1**(7), **EVP_KDF-X942-CONCAT**(7), **EVP_KDF-X963**(7)

HISTORY

Added in OpenSSL 3.0

COPYRIGHT