NAME

openssl-dgst - perform digest operations

SYNOPSIS

openssl dgst|*digest* [**-*digest***] [**-list**] [**-help**] [**-c**] [**-d**] [**-debug**] [**-hex**] [**-binary**] [**-xoflen** *length*] [**-r**] [**-out** *filename*] [**-sign** *filename*|*uri*] [**-keyform DER**|**PEM**|**P12**|**ENGINE**] [**-passin** *arg*] [**-verify** *filename*] [**-prverify** *filename*] [**-signature** *filename*] [**-sigopt** *nm*:*v*] [**-hmac** *key*] [**-mac** *alg*] [**-macopt** *nm*:*v*] [**-fips-fingerprint**] [**-engine** *id*] [**-engine_impl** *id*] [**-rand** *files*] [**-writerand** *file*] [**-provider** *name*] [**-provider-path** *path*] [**-propquery** *propq*] [*file* ...]

DESCRIPTION

This command output the message digest of a supplied file or files in hexadecimal, and also generates and verifies digital signatures using message digests.

The generic name, **openssl dgst**, may be used with an option specifying the algorithm to be used. The default digest is **sha256**. A supported *digest* name may also be used as the sub-command name. To see the list of supported algorithms, use "openssl list -digest-algorithms"

OPTIONS

-help

Print out a usage message.

-digest

Specifies name of a supported digest to be used. See option **-list** below :

-list Prints out a list of supported message digests.

-c Print out the digest in two digit groups separated by colons, only relevant if the **-hex** option is given as well.

-d, **-debug**

Print out BIO debugging information.

-hex

Digest is to be output as a hex dump. This is the default case for a "normal" digest as opposed to a digital signature. See NOTES below for digital signatures using **-hex**.

-binary

Output the digest or signature in binary form.

-xoflen *length*

Set the output length for XOF algorithms, such as **shake128** and **shake256**. This option is not supported for signing operations.

For OpenSSL providers it is recommended to set this value for shake algorithms, since the default values are set to only supply half of the maximum security strength.

For backwards compatibility reasons the default xoflen length for **shake128** is 16 (bytes) which results in a security strength of only 64 bits. To ensure the maximum security strength of 128 bits, the xoflen should be set to at least 32.

For backwards compatibility reasons the default xoflen length for **shake256** is 32 (bytes) which results in a security strength of only 128 bits. To ensure the maximum security strength of 256 bits, the xoflen should be set to at least 64.

-r Output the digest in the "coreutils" format, including newlines. Used by programs like **sha1sum**(1).

-out *filename*

Filename to output to, or standard output by default.

-sign *filename*|*uri*

Digitally sign the digest using the given private key. Note this option does not support Ed25519 or Ed448 private keys. Use the **openssl-pkeyutl**(1) command instead for this.

-keyform DER|**PEM**|**P12**|**ENGINE**

The format of the key to sign with; unspecified by default. See **openssl-format-options**(1) for details.

-sigopt *nm*:*v*

Pass options to the signature algorithm during sign or verify operations. Names and values of these options are algorithm-specific.

-passin *arg*

The private key password source. For more information about the format of *arg* see **openssl-passphrase-options**(1).

-verify *filename*

Verify the signature using the public key in "filename". The output is either "Verified OK" or "Verification Failure".

-prverify *filename*
Verify the signature using the private key in "filename".

-signature *filename*
The actual signature to verify.

-hmac *key*
Create a hashed MAC using "key".

The **openssl-mac**(1) command should be preferred to using this command line option.

-mac *alg*
Create MAC (keyed Message Authentication Code). The most popular MAC algorithm is HMAC (hash-based MAC), but there are other MAC algorithms which are not based on hash, for instance **gost-mac** algorithm, supported by the **gost** engine. MAC keys and other options should be set via **-macopt** parameter.

The **openssl-mac**(1) command should be preferred to using this command line option.

-macopt *nm*:*v*
Passes options to MAC algorithm, specified by **-mac** key. Following options are supported by both by **HMAC** and **gost-mac**:

key:*string*
Specifies MAC key as alphanumeric string (use if key contain printable characters only). String length must conform to any restrictions of the MAC algorithm for example exactly 32 chars for gost-mac.

hexkey:*string*
Specifies MAC key in hexadecimal form (two hex digits per byte). Key length must conform to any restrictions of the MAC algorithm for example exactly 32 chars for gost-mac.

The **openssl-mac**(1) command should be preferred to using this command line option.

-fips-fingerprint
Compute HMAC using a specific key for certain OpenSSL-FIPS operations.

-rand *files*, **-writerand** *file*
See "Random State Options" in **openssl**(1) for details.

-engine *id*

See "Engine Options" in **openssl**(1). This option is deprecated.

The engine is not used for digests unless the **-engine_impl** option is used or it is configured to do so, see "Engine Configuration Module" in **config**(5).

-engine_impl *id*

When used with the **-engine** option, it specifies to also use engine *id* for digest operations.

-provider *name*
-provider-path *path*
-propquery *propq*

See "Provider Options" in **openssl**(1), **provider**(7), and **property**(7).

file ...

File or files to digest. If no files are specified then standard input is used.

EXAMPLES

To create a hex-encoded message digest of a file:

```
openssl dgst -md5 -hex file.txt
or
openssl md5 file.txt
```

To sign a file using SHA-256 with binary file output:

```
openssl dgst -sha256 -sign privatekey.pem -out signature.sign file.txt
or
openssl sha256 -sign privatekey.pem -out signature.sign file.txt
```

To verify a signature:

```
openssl dgst -sha256 -verify publickey.pem \
-signature signature.sign \
file.txt
```

NOTES

The digest mechanisms that are available will depend on the options used when building OpenSSL. The "openssl list -digest-algorithms" command can be used to list them.

New or agile applications should use probably use SHA-256. Other digests, particularly SHA-1 and MD5, are still widely used for interoperating with existing formats and protocols.

When signing a file, this command will automatically determine the algorithm (RSA, ECC, etc) to use for signing based on the private key's ASN.1 info. When verifying signatures, it only handles the RSA, DSA, or ECDSA signature itself, not the related data to identify the signer and algorithm used in formats such as x.509, CMS, and S/MIME.

A source of random numbers is required for certain signing algorithms, in particular ECDSA and DSA.

The signing and verify options should only be used if a single file is being signed or verified.

Hex signatures cannot be verified using **openssl**. Instead, use "xxd -r" or similar program to transform the hex signature into a binary signature prior to verification.

The **openssl-mac**(1) command is preferred over the **-hmac**, **-mac** and **-macopt** command line options.

SEE ALSO

openssl-mac(1)

HISTORY

The default digest was changed from MD5 to SHA256 in OpenSSL 1.1.0. The FIPS-related options were removed in OpenSSL 1.1.0.

The **-engine** and **-engine_impl** options were deprecated in OpenSSL 3.0.

COPYRIGHT