NAME

openssl-glossary - An OpenSSL Glossary

DESCRIPTION

Algorithm

Cryptographic primitives such as the SHA256 digest, or AES encryption are referred to in OpenSSL as "algorithms". There can be more than one implementation for any given algorithm available for use.

crypto(7)

ASN.1, ASN1

ASN.1 ("Abstract Syntax Notation One") is a notation for describing abstract types and values. It is defined in the ITU-T documents X.680 to X.683:

<https://www.itu.int/rec/T-REC-X.680>, <https://www.itu.int/rec/T-REC-X.681>, <https://www.itu.int/rec/T-REC-X.682>, <https://www.itu.int/rec/T-REC-X.683>

Base Provider

An OpenSSL Provider that contains encoders and decoders for OpenSSL keys. All the algorithm implementations in the Base Provider are also available in the Default Provider.

OSSL_PROVIDER-base(7)

Decoder

A decoder is a type of algorithm used for decoding keys and parameters from some external format such as PEM or DER.

OSSL_DECODER_CTX_new_for_pkey(3)

Default Provider

An OpenSSL Provider that contains the most common OpenSSL algorithm implementations. It is loaded by default if no other provider is available. All the algorithm implementations in the Base Provider are also available in the Default Provider.

OSSL_PROVIDER-default(7)

DER ("Distinguished Encoding Rules")

DER is a binary encoding of data, structured according to an ASN.1 specification. This is a common encoding used for cryptographic objects such as private and public keys, certificates,

CRLs, ...

It is defined in ITU-T document X.690:

<https://www.itu.int/rec/T-REC-X.690>

Encoder
: An encoder is a type of algorithm used for encoding keys and parameters to some external format such as PEM or DER.

OSSL_ENCODER_CTX_new_for_pkey(3)

Explicit Fetching
: Explicit Fetching is a type of Fetching (see Fetching). Explicit Fetching is where a function call is made to obtain an algorithm object representing an implementation such as **EVP_MD_fetch**(3) or **EVP_CIPHER_fetch**(3)

Fetching
: Fetching is the process of looking through the available algorithm implementations, applying selection criteria (via a property query string), and finally choosing the implementation that will be used.

Also see Explicit Fetching and Implicit Fetching.

crypto(7)

FIPS Provider
: An OpenSSL Provider that contains OpenSSL algorithm implementations that have been validated according to the FIPS 140-2 standard.

OSSL_PROVIDER-FIPS(7)

Implicit Fetching
: Implicit Fetching is a type of Fetching (see Fetching). Implicit Fetching is where an algorithm object with no associated implementation is used such as the return value from **EVP_sha256**(3) or **EVP_aes_128_cbc**(3). With implicit fetching an implementation is fetched automatically using default selection criteria the first time the algorithm is used.

Legacy Provider
: An OpenSSL Provider that contains algorithm implementations that are considered insecure or are

no longer in common use.

OSSL_PROVIDER-legacy(7)

Library Context

A Library Context in OpenSSL is represented by the type **OSSL_LIB_CTX**. It can be thought of as a scope within which configuration options apply. If an application does not explicitly create a library context then the "default" one is used. Many OpenSSL functions can take a library context as an argument. A NULL value can always be passed to indicate the default library context.

OSSL_LIB_CTX(3)

MSBLOB

MSBLOB is a Microsoft specific binary format for RSA and DSA keys, both private and public. This form is never passphrase protected.

Null Provider

An OpenSSL Provider that contains no algorithm implementations. This can be useful to prevent the default provider from being automatically loaded in a library context.

OSSL_PROVIDER-null(7)

Operation

An operation is a group of OpenSSL functions with a common purpose such as encryption, or digesting.

crypto(7)

PEM ("Privacy Enhanced Message")

PEM is a format used for encoding of binary content into a mail and ASCII friendly form. The content is a series of base64-encoded lines, surrounded by begin/end markers each on their own line. For example:

```
-----BEGIN PRIVATE KEY-----
MIICdg....
... bhTQ==
-----END PRIVATE KEY-----
```

Optional header line(s) may appear after the begin line, and their existence depends on the type of object being written or read.

For all OpenSSL uses, the binary content is expected to be a DER encoded structure.

This is defined in IETF RFC 1421:

<https://tools.ietf.org/html/rfc1421>

PKCS#8

PKCS#8 is a specification of ASN.1 structures that OpenSSL uses for storing or transmitting any private key in a key type agnostic manner. There are two structures worth noting for OpenSSL use, one that contains the key data in unencrypted form (known as "PrivateKeyInfo") and an encrypted wrapper structure (known as "EncryptedPrivateKeyInfo").

This is specified in RFC 5208:

<https://tools.ietf.org/html/rfc5208>

Property

A property is a way of classifying and selecting algorithm implementations. A property is a key/value pair expressed as a string. For example all algorithm implementations in the default provider have the property "provider=default". An algorithm implementation can have multiple properties defined against it.

Also see Property Query String.

property(7)

Property Query String

A property query string is a string containing a sequence of properties that can be used to select an algorithm implementation. For example the query string "provider=example,foo=bar" will select algorithms from the "example" provider that have a "foo" property defined for them with a value of "bar".

Property Query Strings are used during fetching. See Fetching.

property(7)

Provider

A provider in OpenSSL is a component that groups together algorithm implementations. Providers can come from OpenSSL itself or from third parties.

provider(7)

PVK
: PVK is a Microsoft specific binary format for RSA and DSA private keys. This form may be passphrase protected.

SubjectPublicKeyInfo
: SubjectPublicKeyInfo is an ASN.1 structure that OpenSSL uses for storing and transmitting any public key in a key type agnostic manner.

  This is specified as part of the specification for certificates, RFC 5280:

  <https://tools.ietf.org/html/rfc5280>

HISTORY

This glossary was added in OpenSSL 3.0.

COPYRIGHT