NAME

PKCS12_PBE_keyivgen, PKCS12_PBE_keyivgen_ex, PKCS12_pbe_crypt, PKCS12_pbe_crypt_ex - PKCS#12 Password based encryption

SYNOPSIS

```
#include <openssl/evp.h>

int PKCS12_PBE_keyivgen(EVP_CIPHER_CTX *ctx, const char *pass, int passlen,
                        ASN1_TYPE *param, const EVP_CIPHER *cipher,
                        const EVP_MD *md_type, int en_de);
int PKCS12_PBE_keyivgen_ex(EVP_CIPHER_CTX *ctx, const char *pass, int passlen,
                           ASN1_TYPE *param, const EVP_CIPHER *cipher,
                           const EVP_MD *md_type, int en_de,
                           OSSL_LIB_CTX *libctx, const char *propq);
unsigned char *PKCS12_pbe_crypt(const X509_ALGOR *algor,
                                const char *pass, int passlen,
                                const unsigned char *in, int inlen,
                                unsigned char **data, int *datalen,
                                int en_de);
unsigned char *PKCS12_pbe_crypt_ex(const X509_ALGOR *algor,
                                   const char *pass, int passlen,
                                   const unsigned char *in, int inlen,
                                   unsigned char **data, int *datalen,
                                   int en_de, OSSL_LIB_CTX *libctx,
                                   const char *propq);
```

DESCRIPTION

PKCS12_PBE_keyivgen() and **PKCS12_PBE_keyivgen_ex()** take a password *pass* of length *passlen*, parameters *param* and a message digest function *md_type* and perform a key derivation according to PKCS#12. The resulting key is then used to initialise the cipher context *ctx* with a cipher *cipher* for encryption (*en_de*=1) or decryption (*en_de*=0).

PKCS12_PBE_keyivgen_ex() also allows the application to specify a library context *libctx* and property query *propq* to select appropriate algorithm implementations.

PKCS12_pbe_crypt() and **PKCS12_pbe_crypt_ex()** will encrypt or decrypt a buffer based on the algorithm in *algor* and password *pass* of length *passlen*. The input is from *in* of length *inlen* and output is into a malloc'd buffer returned in **data* of length *datalen*. The operation is determined by *en_de*, encryption (*en_de*=1) or decryption (*en_de*=0).

PKCS12_pbe_crypt_ex() allows the application to specify a library context *libctx* and property query *propq* to select appropriate algorithm implementations.

pass is the password used in the derivation of length *passlen*. *pass* is an optional parameter and can be NULL. If *passlen* is -1, then the function will calculate the length of *pass* using **strlen()**.

salt is the salt used in the derivation of length *saltlen*. If the *salt* is NULL, then *saltlen* must be 0. The function will not attempt to calculate the length of the *salt* because it is not assumed to be NULL terminated.

iter is the iteration count and its value should be greater than or equal to 1. RFC 2898 suggests an iteration count of at least 1000. Any *iter* less than 1 is treated as a single iteration.

digest is the message digest function used in the derivation.

Functions ending in **_ex()** take optional parameters *libctx* and *propq* which are used to select appropriate algorithm implementations.

NOTES

The functions are typically used in PKCS#12 to encrypt objects.

These functions make no assumption regarding the given password. It will simply be treated as a byte sequence.

RETURN VALUES

PKCS12_PBE_keyivgen(), **PKCS12_PBE_keyivgen_ex()** return 1 on success or 0 on error.

PKCS12_pbe_crypt() and **PKCS12_pbe_crypt_ex()** return a buffer containing the output or NULL if an error occurred.

CONFORMING TO

IETF RFC 7292 (<https://tools.ietf.org/html/rfc7292>)

SEE ALSO

EVP_PBE_CipherInit_ex(3), **PKCS8_encrypt_ex**(3), **passphrase-encoding**(7)

HISTORY

PKCS12_PBE_keyivgen_ex() and **PKCS12_pbe_crypt_ex()** were added in OpenSSL 3.0.

COPYRIGHT