NAME

EVP_PKEY_CTX_set1_scrypt_salt, EVP_PKEY_CTX_set_scrypt_N,
EVP_PKEY_CTX_set_scrypt_r, EVP_PKEY_CTX_set_scrypt_p,
EVP_PKEY_CTX_set_scrypt_maxmem_bytes - EVP_PKEY scrypt KDF support functions

SYNOPSIS

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

int EVP_PKEY_CTX_set1_scrypt_salt(EVP_PKEY_CTX *pctx, unsigned char *salt,
                                  int saltlen);

int EVP_PKEY_CTX_set_scrypt_N(EVP_PKEY_CTX *pctx, uint64_t N);

int EVP_PKEY_CTX_set_scrypt_r(EVP_PKEY_CTX *pctx, uint64_t r);

int EVP_PKEY_CTX_set_scrypt_p(EVP_PKEY_CTX *pctx, uint64_t p);

int EVP_PKEY_CTX_set_scrypt_maxmem_bytes(EVP_PKEY_CTX *pctx,
                                         uint64_t maxmem);
```

DESCRIPTION

These functions are used to set up the necessary data to use the scrypt KDF. For more information on scrypt, see **EVP_KDF-SCRYPT**(7).

EVP_PKEY_CTX_set1_scrypt_salt() sets the **saltlen** bytes long salt value.

EVP_PKEY_CTX_set_scrypt_N(), **EVP_PKEY_CTX_set_scrypt_r()** and **EVP_PKEY_CTX_set_scrypt_p()** configure the work factors N, r and p.

EVP_PKEY_CTX_set_scrypt_maxmem_bytes() sets how much RAM key derivation may maximally use, given in bytes. If RAM is exceeded because the load factors are chosen too high, the key derivation will fail.

STRING CTRLS

scrypt also supports string based control operations via **EVP_PKEY_CTX_ctrl_str**(3). Similarly, the **salt** can either be specified using the **type** parameter "salt" or in hex encoding by using the "hexsalt" parameter. The work factors **N**, **r** and **p** as well as **maxmem_bytes** can be set by using the parameters "N", "r", "p" and "maxmem_bytes", respectively.

NOTES

There is a newer generic API for KDFs, **EVP_KDF**(3), which is preferred over the EVP_PKEY method.

The scrypt KDF also uses **EVP_PKEY_CTX_set1_pbe_pass()** as well as the value from the string controls "pass" and "hexpass". See **EVP_PKEY_CTX_set1_pbe_pass**(3).

RETURN VALUES

All these functions return 1 for success and 0 or a negative value for failure. In particular a return value of -2 indicates the operation is not supported by the public key algorithm.

SEE ALSO

EVP_KDF(3) **EVP_PKEY_CTX_new**(3), **EVP_PKEY_CTX_ctrl_str**(3), **EVP_PKEY_derive**(3)

HISTORY

All of the functions described here were converted from macros to functions in OpenSSL 3.0.

COPYRIGHT