Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'RSA_generate_key_ex.3ossl' command

$ man RSA_generate_key_ex.3ossl

RSA_GENERATE_KEY(3ossl) OpenSSL RSA_GENERATE_KEY(3ossl)

NAME

EVP_RSA_gen, RSA_generate_key_ex, RSA_generate_key,

RSA_generate_multi_prime_key - generate RSA key pair

SYNOPSIS

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

EVP_PKEY *EVP_RSA_gen(unsigned int bits);
```

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining OPENSSL_API_COMPAT with a suitable version value, see openssl_user_macros(7):

```
int RSA_generate_key_ex(RSA *rsa, int bits, BIGNUM *e, BN_GENCB *cb);
int RSA_generate_multi_prime_key(RSA *rsa, int bits, int primes, BIGNUM *e, BN_GENCB *cb);
```

The following function has been deprecated since OpenSSL 0.9.8, and can be hidden entirely by defining OPENSSL_API_COMPAT with a suitable version value, see openssl_user_macros(7):

```
RSA *RSA_generate_key(int bits, unsigned long e,
```

```
void (*callback)(int, int, void *), void *cb_arg);
```

DESCRIPTION

EVP_RSA_gen() generates a new RSA key pair with modulus size bits.

All of the functions described below are deprecated. Applications should instead use EVP_RSA_gen(), EVP_PKEY_Q_keygen(3), or EVP_PKEY_keygen_init(3) and EVP_PKEY_keygen(3).

RSA_generate_key_ex() generates a 2-prime RSA key pair and stores it in the RSA structure provided in rsa.

RSA_generate_multi_prime_key() generates a multi-prime RSA key pair and stores it in the RSA structure provided in rsa. The number of primes is given by the primes parameter. If the automatic seeding or reseeding of the OpenSSL CSPRNG fails due to external circumstances (see RAND(7)), the operation will fail.

The modulus size will be of length bits, the number of primes to form the modulus will be primes, and the public exponent will be e. Key sizes with num < 1024 should be considered insecure. The exponent is an odd number, typically 3, 17 or 65537.

In order to maintain adequate security level, the maximum number of permitted primes depends on modulus bit length:

```
  <1024 | >=1024 | >=4096 | >=8192
  ------+--------+--------+-------
    2   |   3    |   4    |   5
```

A callback function may be used to provide feedback about the progress of the key generation. If cb is not NULL, it will be called as follows using the BN_GENCB_call() function described on the

BN_generate_prime(3) page.

RSA_generate_key() is similar to RSA_generate_key_ex() but expects an old-style callback function; see BN_generate_prime(3) for information on the old-style callback.

? While a random prime number is generated, it is called as described in BN_generate_prime(3).

? When the n-th randomly generated prime is rejected as not suitable for the key, BN_GENCB_call(cb, 2, n) is called.

? When a random p has been found with p-1 relatively prime to e, it is called as BN_GENCB_call(cb, 3, 0).

The process is then repeated for prime q and other primes (if any) with BN_GENCB_call(cb, 3, i) where i indicates the i-th prime.

RETURN VALUES

EVP_RSA_gen() returns an EVP_PKEY or NULL on failure.

RSA_generate_multi_prime_key() returns 1 on success or 0 on error. RSA_generate_key_ex() returns 1 on success or 0 on error. The error codes can be obtained by ERR_get_error(3).

RSA_generate_key() returns a pointer to the RSA structure or NULL if the key generation fails.

BUGS

BN_GENCB_call(cb, 2, x) is used with two different meanings.

SEE ALSO

EVP_PKEY_Q_keygen(3) BN_generate_prime(3), ERR_get_error(3),

RAND_bytes(3), RAND(7)

HISTORY

EVP_RSA_gen() was added in OpenSSL 3.0. All other functions described here were deprecated in OpenSSL 3.0. For replacement see EVP_PKEY-RSA(7).

COPYRIGHT