

**NAME**

BN\_rand, BN\_pseudo\_rand, BN\_rand\_range, BN\_pseudo\_rand\_range - generate pseudo-random number

**SYNOPSIS**

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

int BN_rand(BIGNUM *rnd, int bits, int top, int bottom);

int BN_pseudo_rand(BIGNUM *rnd, int bits, int top, int bottom);

int BN_rand_range(BIGNUM *rnd, BIGNUM *range);

int BN_pseudo_rand_range(BIGNUM *rnd, BIGNUM *range);
```

**DESCRIPTION**

*BN\_rand()* generates a cryptographically strong pseudo-random number of **bits** bits in length and stores it in **rnd**. If **top** is -1, the most significant bit of the random number can be zero. If **top** is 0, it is set to 1, and if **top** is 1, the two most significant bits of the number will be set to 1, so that the product of two such random numbers will always have  $2*\text{bits}$  length. If **bottom** is true, the number will be odd.

*BN\_pseudo\_rand()* does the same, but pseudo-random numbers generated by this function are not necessarily unpredictable. They can be used for non-cryptographic purposes and for certain purposes in cryptographic protocols, but usually not for key generation etc.

*BN\_rand\_range()* generates a cryptographically strong pseudo-random number **rnd** in the range  $0 < \text{rnd} \leq \text{range}$ . *BN\_pseudo\_rand\_range()* does the same, but is based on *BN\_pseudo\_rand()*, and hence numbers generated by it are not necessarily unpredictable.

The PRNG must be seeded prior to calling *BN\_rand()* or *BN\_rand\_range()*.

**RETURN VALUES**

The functions return 1 on success, 0 on error. The error codes can be obtained by [ERR\\_get\\_error\(3\)](#).

**SEE ALSO**

[bn\(3\)](#), [ERR\\_get\\_error\(3\)](#), [rand\(3\)](#), [RAND\\_add\(3\)](#), [RAND\\_bytes\(3\)](#)

**HISTORY**

*BN\_rand()* is available in all versions of SSLeay and OpenSSL. *BN\_pseudo\_rand()* was added in OpenSSL 0.9.5. The **top** == -1 case and the function *BN\_rand\_range()* were added in OpenSSL 0.9.6a. *BN\_pseudo\_rand\_range()* was added in OpenSSL 0.9.6c.