

**NAME**

BN\_mod\_mul\_reciprocal, BN\_div\_recip, BN\_RECP\_CTX\_new, BN\_RECP\_CTX\_init,  
BN\_RECP\_CTX\_free, BN\_RECP\_CTX\_set - modular multiplication using reciprocal

**SYNOPSIS**

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

BN_RECP_CTX *BN_RECP_CTX_new(void);
void BN_RECP_CTX_init(BN_RECP_CTX *recp);
void BN_RECP_CTX_free(BN_RECP_CTX *recp);

int BN_RECP_CTX_set(BN_RECP_CTX *recp, const BIGNUM *m, BN_CTX *ctx);

int BN_div_recip(BIGNUM *dv, BIGNUM *rem, BIGNUM *a, BN_RECP_CTX *recp,
BN_CTX *ctx);

int BN_mod_mul_reciprocal(BIGNUM *r, BIGNUM *a, BIGNUM *b,
BN_RECP_CTX *recp, BN_CTX *ctx);
```

**DESCRIPTION**

*BN\_mod\_mul\_reciprocal()* can be used to perform an efficient *BN\_mod\_mul(3)* operation when the operation will be performed repeatedly with the same modulus. It computes  $r=(a*b)\%m$  using  $recp=1/m$ , which is set as described below. *ctx* is a previously allocated **BN\_CTX** used for temporary variables.

*BN\_RECP\_CTX\_new()* allocates and initializes a **BN\_RECP** structure. *BN\_RECP\_CTX\_init()* initializes an existing uninitialized **BN\_RECP**.

*BN\_RECP\_CTX\_free()* frees the components of the **BN\_RECP**, and, if it was created by *BN\_RECP\_CTX\_new()*, also the structure itself.

*BN\_RECP\_CTX\_set()* stores *m* in *recp* and sets it up for computing  $1/m$  and shifting it left by  $BN\_num\_bits(m)+1$  to make it an integer. The result and the number of bits it was shifted left will later be stored in *recp*.

*BN\_div\_recip()* divides *a* by *m* using *recp*. It places the quotient in *dv* and the remainder in *rem*.

The **BN\_RECP\_CTX** structure is defined as follows:

```
typedef struct bn_recp_ctx_st
{
    BIGNUM N; /* the divisor */
    BIGNUM Nr; /* the reciprocal */
    int num_bits;
    int shift;
    int flags;
} BN_RECP_CTX;
```

It cannot be shared between threads.

**RETURN VALUES**

*BN\_RECP\_CTX\_new()* returns the newly allocated **BN\_RECP\_CTX**, and NULL on error.

*BN\_RECP\_CTX\_init()* and *BN\_RECP\_CTX\_free()* have no return values.

For the other functions, 1 is returned for success, 0 on error. The error codes can be obtained by *ERR\_get\_error(3)*.

**SEE ALSO**

*bn(3)*, *ERR\_get\_error(3)*, *BN\_add(3)*, *BN\_CTX\_new(3)*

**HISTORY**

**BN\_RECP\_CTX** was added in SSLeay 0.9.0. Before that, the function *BN\_reciprocal()* was used instead, and the *BN\_mod\_mul\_reciprocal()* arguments were different.