

**NAME**

BN\_BLINDING\_new, BN\_BLINDING\_free, BN\_BLINDING\_update, BN\_BLINDING\_convert, BN\_BLINDING\_invert, BN\_BLINDING\_convert\_ex, BN\_BLINDING\_invert\_ex, BN\_BLINDING\_get\_thread\_id, BN\_BLINDING\_set\_thread\_id, BN\_BLINDING\_thread\_id, BN\_BLINDING\_get\_flags, BN\_BLINDING\_set\_flags, BN\_BLINDING\_create\_param - blinding related BIGNUM functions.

**SYNOPSIS**

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

BN_BLINDING *BN_BLINDING_new(const BIGNUM *A, const BIGNUM *Ai,
    BIGNUM *mod);
void BN_BLINDING_free(BN_BLINDING *b);
int BN_BLINDING_update(BN_BLINDING *b, BN_CTX *ctx);
int BN_BLINDING_convert(BIGNUM *n, BN_BLINDING *b, BN_CTX *ctx);
int BN_BLINDING_invert(BIGNUM *n, BN_BLINDING *b, BN_CTX *ctx);
int BN_BLINDING_convert_ex(BIGNUM *n, BIGNUM *r, BN_BLINDING *b,
    BN_CTX *ctx);
int BN_BLINDING_invert_ex(BIGNUM *n, const BIGNUM *r, BN_BLINDING *b,
    BN_CTX *ctx);
#ifdef OPENSSL_NO_DEPRECATED
unsigned long BN_BLINDING_get_thread_id(const BN_BLINDING *);
void BN_BLINDING_set_thread_id(BN_BLINDING *, unsigned long);
#endif
CRYPTO_THREADID *BN_BLINDING_thread_id(BN_BLINDING *);
unsigned long BN_BLINDING_get_flags(const BN_BLINDING *);
void BN_BLINDING_set_flags(BN_BLINDING *, unsigned long);
BN_BLINDING *BN_BLINDING_create_param(BN_BLINDING *b,
    const BIGNUM *e, BIGNUM *m, BN_CTX *ctx,
    int (*bn_mod_exp)(BIGNUM *r, const BIGNUM *a, const BIGNUM *p,
    const BIGNUM *m, BN_CTX *ctx, BN_MONT_CTX *m_ctx),
    BN_MONT_CTX *m_ctx);
```

**DESCRIPTION**

*BN\_BLINDING\_new()* allocates a new **BN\_BLINDING** structure and copies the **A** and **Ai** values into the newly created **BN\_BLINDING** object.

*BN\_BLINDING\_free()* frees the **BN\_BLINDING** structure.

*BN\_BLINDING\_update()* updates the **BN\_BLINDING** parameters by squaring the **A** and **Ai** or, after specific number of uses and if the necessary parameters are set, by re-creating the blinding parameters.

*BN\_BLINDING\_convert\_ex()* multiplies **n** with the blinding factor **A**. If **r** is not NULL a copy of the inverse blinding factor **Ai** will be returned in **r** (this is useful if a **RSA** object is shared among several threads).

*BN\_BLINDING\_invert\_ex()* multiplies **n** with the inverse blinding factor **Ai**. If **r** is not NULL it will be used as the inverse blinding.

*BN\_BLINDING\_convert()* and *BN\_BLINDING\_invert()* are wrapper functions for *BN\_BLINDING\_convert\_ex()* and *BN\_BLINDING\_invert\_ex()* with **r** set to NULL.

*BN\_BLINDING\_thread\_id()* provides access to the **CRYPTO\_THREADID** object within the **BN\_BLINDING** structure. This is to help users provide proper locking if needed for multi-threaded use. The “thread id” object of a newly allocated **BN\_BLINDING** structure is initialised to the thread id in which *BN\_BLINDING\_new()* was called.

*BN\_BLINDING\_get\_flags()* returns the **BN\_BLINDING** flags. Currently there are two supported flags: **BN\_BLINDING\_NO\_UPDATE** and **BN\_BLINDING\_NO\_RECREATE**. **BN\_BLINDING\_NO\_UPDATE** inhibits the automatic update of the **BN\_BLINDING** parameters after each use and **BN\_BLINDING\_NO\_RECREATE** inhibits the automatic re-creation of the **BN\_BLINDING** parameters after

a fixed number of uses (currently 32). In newly allocated **BN\_BLINDING** objects no flags are set. *BN\_BLINDING\_set\_flags()* sets the **BN\_BLINDING** parameters flags.

*BN\_BLINDING\_create\_param()* creates new **BN\_BLINDING** parameters using the exponent **e** and the modulus **m**. **bn\_mod\_exp** and **m\_ctx** can be used to pass special functions for exponentiation (normally *BN\_mod\_exp\_mont()* and **BN\_MONT\_CTX**).

## RETURN VALUES

*BN\_BLINDING\_new()* returns the newly allocated **BN\_BLINDING** structure or NULL in case of an error.

*BN\_BLINDING\_update()*, *BN\_BLINDING\_convert()*, *BN\_BLINDING\_invert()*,  
*BN\_BLINDING\_convert\_ex()* and *BN\_BLINDING\_invert\_ex()* return 1 on success and 0 if an error occurred.

*BN\_BLINDING\_thread\_id()* returns a pointer to the thread id object within a **BN\_BLINDING** object.

*BN\_BLINDING\_get\_flags()* returns the currently set **BN\_BLINDING** flags (a **unsigned long** value).

*BN\_BLINDING\_create\_param()* returns the newly created **BN\_BLINDING** parameters or NULL on error.

## SEE ALSO

[bn\(3\)](#)

## HISTORY

**BN\_BLINDING\_thread\_id** was first introduced in OpenSSL 1.0.0, and it deprecates **BN\_BLINDING\_set\_thread\_id** and **BN\_BLINDING\_get\_thread\_id**.

**BN\_BLINDING\_convert\_ex**, **BN\_BLINDING\_invert\_ex**, **BN\_BLINDING\_get\_thread\_id**,  
**BN\_BLINDING\_set\_thread\_id**, **BN\_BLINDING\_set\_flags**, **BN\_BLINDING\_get\_flags** and  
**BN\_BLINDING\_create\_param** were first introduced in OpenSSL 0.9.8

## AUTHOR

Nils Larsch for the OpenSSL project (<http://www.openssl.org>).