

NAME

BN_num_bits, BN_num_bytes, BN_num_bits_word - get BIGNUM size

SYNOPSIS

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

int BN_num_bytes(const BIGNUM *a);

int BN_num_bits(const BIGNUM *a);

int BN_num_bits_word(BN_ULONG w);
```

DESCRIPTION

BN_num_bytes() returns the size of a **BIGNUM** in bytes.

BN_num_bits_word() returns the number of significant bits in a word. If we take 0x00000432 as an example, it returns 11, not 16, not 32. Basically, except for a zero, it returns $\text{floor}(\log_2(w))+1$.

BN_num_bits() returns the number of significant bits in a **BIGNUM**, following the same principle as *BN_num_bits_word()*.

BN_num_bytes() is a macro.

RETURN VALUES

The size.

NOTES

Some have tried using *BN_num_bits()* on individual numbers in RSA keys, DH keys and DSA keys, and found that they don't always come up with the number of bits they expected (something like 512, 1024, 2048, ...). This is because generating a number with some specific number of bits doesn't always set the highest bits, thereby making the number of *significant* bits a little lower. If you want to know the "key size" of such a key, either use functions like *RSA_size()*, *DH_size()* and *DSA_size()*, or use *BN_num_bytes()* and multiply with 8 (although there's no real guarantee that will match the "key size", just a lot more probability).

SEE ALSO

[bn\(3\)](#), [DH_size\(3\)](#), [DSA_size\(3\)](#), [RSA_size\(3\)](#)

HISTORY

BN_num_bytes(), *BN_num_bits()* and *BN_num_bits_word()* are available in all versions of SSLeay and OpenSSL.