NAME

EVP_PKEY_get_size, EVP_PKEY_get_bits, EVP_PKEY_get_security_bits, EVP_PKEY_bits, EVP_PKEY_security_bits, EVP_PKEY_size - EVP_PKEY information functions

SYNOPSIS

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

int EVP_PKEY_get_size(const EVP_PKEY *pkey);
int EVP_PKEY_get_bits(const EVP_PKEY *pkey);
int EVP_PKEY_get_security_bits(const EVP_PKEY *pkey);

#define EVP_PKEY_bits EVP_PKEY_get_bits
#define EVP_PKEY_security_bits EVP_PKEY_get_security_bits
#define EVP_PKEY_size EVP_PKEY_get_size
```

DESCRIPTION

EVP_PKEY_get_size() returns the maximum suitable size for the output buffers for almost all operations that can be done with *pkey*. The primary documented use is with **EVP_SignFinal**(3) and **EVP_SealInit**(3), but it isn't limited there. The returned size is also large enough for the output buffer of **EVP_PKEY_sign**(3), **EVP_PKEY_encrypt**(3), **EVP_PKEY_decrypt**(3), **EVP_PKEY_derive**(3).

It must be stressed that, unless the documentation for the operation that's being performed says otherwise, the size returned by **EVP_PKEY_get_size()** is only preliminary and not exact, so the final contents of the target buffer may be smaller. It is therefore crucial to take note of the size given back by the function that performs the operation, such as **EVP_PKEY_sign**(3) (the *siglen* argument will receive that length), to avoid bugs.

EVP_PKEY_get_bits() returns the cryptographic length of the cryptosystem to which the key in *pkey* belongs, in bits. Note that the definition of cryptographic length is specific to the key cryptosystem.

EVP_PKEY_get_security_bits() returns the number of security bits of the given *pkey*, bits of security is defined in NIST SP800-57.

RETURN VALUES

EVP_PKEY_get_size(), **EVP_PKEY_get_bits()** and **EVP_PKEY_get_security_bits()** return a positive number, or 0 if this size isn't available.

NOTES

Most functions that have an output buffer and are mentioned with **EVP_PKEY_get_size()** have a functionality where you can pass NULL for the buffer and still pass a pointer to an integer and get the

exact size that this function call delivers in the context that it's called in. This allows those functions to be called twice, once to find out the exact buffer size, then allocate the buffer in between, and call that function again actually output the data. For those functions, it isn't strictly necessary to call **EVP_PKEY_get_size()** to find out the buffer size, but may be useful in cases where it's desirable to know the upper limit in advance.

It should also be especially noted that **EVP_PKEY_get_size()** shouldn't be used to get the output size for **EVP_DigestSignFinal()**, according to "NOTES" in **EVP_DigestSignFinal**(3).

SEE ALSO

EVP_SignFinal(3), **EVP_SealInit**(3), **EVP_PKEY_sign**(3), **EVP_PKEY_encrypt**(3), **EVP_PKEY_decrypt**(3), **EVP_PKEY_derive**(3)

HISTORY

The **EVP_PKEY_bits()**, **EVP_PKEY_security_bits()**, and **EVP_PKEY_size()** functions were renamed to include "get" in their names in OpenSSL 3.0, respectively. The old names are kept as non-deprecated alias macros.

COPYRIGHT