

**NAME**

RSA\_check\_key\_ex, RSA\_check\_key - validate private RSA keys

**SYNOPSIS**

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

The following functions have been deprecated since OpenSSL 3.0, and can be hidden entirely by defining **OPENSSL\_API\_COMPAT** with a suitable version value, see **openssl\_user\_macros(7)**:

```
int RSA_check_key_ex(const RSA *rsa, BN_GENCB *cb);
```

```
int RSA_check_key(const RSA *rsa);
```

**DESCRIPTION**

Both of the functions described on this page are deprecated. Applications should instead use **EVP\_PKEY\_public\_check(3)**, **EVP\_PKEY\_private\_check(3)** and **EVP\_PKEY\_pairwise\_check(3)**.

**RSA\_check\_key\_ex()** function validates RSA keys. It checks that **p** and **q** are in fact prime, and that **n** = **p\*q**.

It does not work on RSA public keys that have only the modulus and public exponent elements populated. It also checks that **d\*e = 1 mod (p-1\*q-1)**, and that **dmp1**, **dmq1** and **iqmp** are set correctly or are **NULL**. It performs integrity checks on all the RSA key material, so the RSA key structure must contain all the private key data too. Therefore, it cannot be used with any arbitrary RSA key object, even if it is otherwise fit for regular RSA operation.

The **cb** parameter is a callback that will be invoked in the same manner as **BN\_is\_prime\_ex(3)**.

**RSA\_check\_key()** is equivalent to **RSA\_check\_key\_ex()** with a **NULL cb**.

**RETURN VALUES**

**RSA\_check\_key\_ex()** and **RSA\_check\_key()** return 1 if **rsa** is a valid RSA key, and 0 otherwise. They return -1 if an error occurs while checking the key.

If the key is invalid or an error occurred, the reason code can be obtained using **ERR\_get\_error(3)**.

**NOTES**

Unlike most other RSA functions, this function does **not** work transparently with any underlying **ENGINE** implementation because it uses the key data in the RSA structure directly. An **ENGINE** implementation can override the way key data is stored and handled, and can even provide support for

HSM keys - in which case the RSA structure may contain **no** key data at all! If the ENGINE in question is only being used for acceleration or analysis purposes, then in all likelihood the RSA key data is complete and untouched, but this can't be assumed in the general case.

## BUGS

A method of verifying the RSA key using opaque RSA API functions might need to be considered. Right now **RSA\_check\_key()** simply uses the RSA structure elements directly, bypassing the RSA\_METHOD table altogether (and completely violating encapsulation and object-orientation in the process). The best fix will probably be to introduce a "**check\_key()**" handler to the RSA\_METHOD function table so that alternative implementations can also provide their own verifiers.

## SEE ALSO

**BN\_is\_prime\_ex(3)**, **ERR\_get\_error(3)**

## HISTORY

All of these functions were deprecated in OpenSSL 3.0.

**RSA\_check\_key\_ex()** appeared after OpenSSL 1.0.2.

## COPYRIGHT

Copyright 2000-2021 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.