NAME

RSA_public_encrypt, RSA_private_decrypt - RSA public key cryptography

SYNOPSIS

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

int RSA_public_encrypt(int flen, const unsigned char *from,
unsigned char *to, RSA *rsa, int padding);

int RSA_private_decrypt(int flen, const unsigned char *from,
unsigned char *to, RSA *rsa, int padding);
```

DESCRIPTION

RSA_public_encrypt() encrypts the **flen** bytes at **from** (usually a session key) using the public key **rsa** and stores the ciphertext in **to**. **to** must point to RSA_size(**rsa**) bytes of memory.

padding denotes one of the following modes:

RSA_PKCS1_PADDING
: PKCS #1 v1.5 padding. This currently is the most widely used mode.

RSA_PKCS1_OAEP_PADDING
: EME-OAEP as defined in PKCS #1 v2.0 with SHA-1, MGF1 and an empty encoding parameter. This mode is recommended for all new applications.

RSA_SSLV23_PADDING
: PKCS #1 v1.5 padding with an SSL-specific modification that denotes that the server is SSL3 capable.

RSA_NO_PADDING
: Raw RSA encryption. This mode should *only* be used to implement cryptographically sound padding modes in the application code. Encrypting user data directly with RSA is insecure.

flen must be less than RSA_size(**rsa**) - 11 for the PKCS #1 v1.5 based padding modes, less than RSA_size(**rsa**) - 41 for RSA_PKCS1_OAEP_PADDING and exactly RSA_size(**rsa**) for RSA_NO_PADDING. The random number generator must be seeded prior to calling *RSA_public_encrypt()*.

RSA_private_decrypt() decrypts the **flen** bytes at **from** using the private key **rsa** and stores the plaintext in **to**. **to** must point to a memory section large enough to hold the decrypted data (which is smaller than RSA_size(**rsa**)). **padding** is the padding mode that was used to encrypt the data.

RETURN VALUES

RSA_public_encrypt() returns the size of the encrypted data (i.e., RSA_size(**rsa**)). *RSA_private_decrypt()* returns the size of the recovered plaintext.

On error, -1 is returned; the error codes can be obtained by *ERR_get_error(3)* .

WARNING

Decryption failures in the RSA_PKCS1_PADDING mode leak information which can potentially be used to mount a Bleichenbacher padding oracle attack. This is an inherent weakness in the PKCS #1 v1.5 padding design. Prefer RSA_PKCS1_OAEP_PADDING.

CONFORMING TO

SSL, PKCS #1 v2.0

SEE ALSO

ERR_get_error(3) , *rand(3)* , *RSA_size(3)*

COPYRIGHT