NAME

EVP_PKEY_check, EVP_PKEY_param_check, EVP_PKEY_param_check_quick, EVP_PKEY_public_check, EVP_PKEY_public_check_quick, EVP_PKEY_private_check, EVP_PKEY_pairwise_check - key and parameter validation functions

SYNOPSIS

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

int EVP_PKEY_check(EVP_PKEY_CTX *ctx);
int EVP_PKEY_param_check(EVP_PKEY_CTX *ctx);
int EVP_PKEY_param_check_quick(EVP_PKEY_CTX *ctx);
int EVP_PKEY_public_check(EVP_PKEY_CTX *ctx);
int EVP_PKEY_public_check_quick(EVP_PKEY_CTX *ctx);
int EVP_PKEY_private_check(EVP_PKEY_CTX *ctx);
int EVP_PKEY_pairwise_check(EVP_PKEY_CTX *ctx);
```

DESCRIPTION

EVP_PKEY_param_check() validates the parameters component of the key given by **ctx**. This check will always succeed for key types that do not have parameters.

EVP_PKEY_param_check_quick() validates the parameters component of the key given by **ctx** like **EVP_PKEY_param_check()** does. However some algorithm implementations may offer a quicker form of validation that omits some checks in order to perform a lightweight sanity check of the key. If a quicker form is not provided then this function call does the same thing as **EVP_PKEY_param_check()**.

EVP_PKEY_public_check() validates the public component of the key given by **ctx**.

EVP_PKEY_public_check_quick() validates the public component of the key given by **ctx** like **EVP_PKEY_public_check()** does. However some algorithm implementations may offer a quicker form of validation that omits some checks in order to perform a lightweight sanity check of the key. If a quicker form is not provided then this function call does the same thing as **EVP_PKEY_public_check()**.

EVP_PKEY_private_check() validates the private component of the key given by **ctx**.

EVP_PKEY_pairwise_check() validates that the public and private components have the correct mathematical relationship to each other for the key given by **ctx**.

EVP_PKEY_check() is an alias for the **EVP_PKEY_pairwise_check()** function.

NOTES

Key validation used by the OpenSSL FIPS provider complies with the rules within SP800-56A and SP800-56B. For backwards compatibility reasons the OpenSSL default provider may use checks that are not as restrictive for certain key types. For further information see "DSA key validation" in **EVP_PKEY-DSA**(7), "DH key validation" in **EVP_PKEY-DH**(7), "EC key validation" in **EVP_PKEY-EC**(7) and "RSA key validation" in **EVP_PKEY-RSA**(7).

Refer to SP800-56A and SP800-56B for rules relating to when these functions should be called during key establishment. It is not necessary to call these functions after locally calling an approved key generation method, but may be required for assurance purposes when receiving keys from a third party.

RETURN VALUES

All functions return 1 for success or others for failure. They return -2 if the operation is not supported for the specific algorithm.

SEE ALSO

EVP_PKEY_CTX_new(3), **EVP_PKEY_fromdata**(3), **EVP_PKEY-DH**(7), **EVP_PKEY-FFC**(7), **EVP_PKEY-DSA**(7), **EVP_PKEY-EC**(7), **EVP_PKEY-RSA**(7),

HISTORY

EVP_PKEY_check(), **EVP_PKEY_public_check()** and **EVP_PKEY_param_check()** were added in OpenSSL 1.1.1.

EVP_PKEY_param_check_quick(), **EVP_PKEY_public_check_quick()**, **EVP_PKEY_private_check()** and **EVP_PKEY_pairwise_check()** were added in OpenSSL 3.0.

COPYRIGHT