NAME

RAND_set_rand_method, RAND_get_rand_method, RAND_OpenSSL - select RAND method

SYNOPSIS

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

void RAND_set_rand_method(const RAND_METHOD *meth);

const RAND_METHOD *RAND_get_rand_method(void);

RAND_METHOD *RAND_OpenSSL(void);
```

DESCRIPTION

A **RAND_METHOD** specifies the functions that OpenSSL uses for random number generation. By modifying the method, alternative implementations such as hardware RNGs may be used. IMPORTANT: See the NOTES section for important information about how these RAND API functions are affected by the use of **ENGINE** API calls.

Initially, the default RAND_METHOD is the OpenSSL internal implementation, as returned by *RAND_OpenSSL()*.

RAND_set_default_method() makes **meth** the method for PRNG use. **NB**: This is true only whilst no ENGINE has been set as a default for RAND, so this function is no longer recommended.

RAND_get_default_method() returns a pointer to the current RAND_METHOD. However, the meaningfulness of this result is dependent on whether the ENGINE API is being used, so this function is no longer recommended.

THE RAND_METHOD STRUCTURE

```
typedef struct rand_meth_st
{
void (*seed)(const void *buf, int num);
int (*bytes)(unsigned char *buf, int num);
void (*cleanup)(void);
void (*add)(const void *buf, int num, int entropy);
int (*pseudorand)(unsigned char *buf, int num);
int (*status)(void);
} RAND_METHOD;
```

The components point to method implementations used by (or called by), in order, *RAND_seed()*, *RAND_bytes()*, internal RAND cleanup, *RAND_add()*, *RAND_pseudo_rand()* and *RAND_status()*. Each component may be NULL if the function is not implemented.

RETURN VALUES

RAND_set_rand_method() returns no value. *RAND_get_rand_method()* and *RAND_OpenSSL()* return pointers to the respective methods.

NOTES

RAND_METHOD implementations are grouped together with other algorithmic APIs (eg. RSA_METHOD, EVP_CIPHER, etc) in **ENGINE** modules. If a default ENGINE is specified for RAND functionality using an ENGINE API function, that will override any RAND defaults set using the RAND API (ie. *RAND_set_rand_method()*). For this reason, the ENGINE API is the recommended way to control default implementations for use in RAND and other cryptographic algorithms.

SEE ALSO

rand(3) , *engine(3)*

COPYRIGHT