NAME

EVP_RAND−HMAC−DRBG − The HMAC DRBG EVP_RAND implementation

DESCRIPTION

Support for the HMAC deterministic random bit generator through the **EVP_RAND** API.

Identity

"HMAC-DRBG" is the name for this implementation; it can be used with the **EVP_RAND_fetch()** function.

Supported parameters

The supported parameters are:

"state" (**OSSL_RAND_PARAM_STATE**) <integer>
"strength" (**OSSL_RAND_PARAM_STRENGTH**) <unsigned integer>
"max_request" (**OSSL_RAND_PARAM_MAX_REQUEST**) <unsigned integer>
"reseed_requests" (**OSSL_DRBG_PARAM_RESEED_REQUESTS**) <unsigned integer>
"reseed_time_interval" (**OSSL_DRBG_PARAM_RESEED_TIME_INTERVAL**) <integer>
"min_entropylen" (**OSSL_DRBG_PARAM_MIN_ENTROPYLEN**) <unsigned integer>
"max_entropylen" (**OSSL_DRBG_PARAM_MAX_ENTROPYLEN**) <unsigned integer>
"min_noncelen" (**OSSL_DRBG_PARAM_MIN_NONCELEN**) <unsigned integer>
"max_noncelen" (**OSSL_DRBG_PARAM_MAX_NONCELEN**) <unsigned integer>
"max_perslen" (**OSSL_DRBG_PARAM_MAX_PERSLEN**) <unsigned integer>
"max_adinlen" (**OSSL_DRBG_PARAM_MAX_ADINLEN**) <unsigned integer>
"reseed_counter" (**OSSL_DRBG_PARAM_RESEED_COUNTER**) <unsigned integer>
"properties" (**OSSL_DRBG_PARAM_PROPERTIES**) <UTF8 string>
"mac" (**OSSL_DRBG_PARAM_MAC**) <UTF8 string>
"digest" (**OSSL_DRBG_PARAM_DIGEST**) <UTF8 string>

These parameters work as described in "PARAMETERS" in **EVP_RAND** (3).

NOTES

A context for HMAC DRBG can be obtained by calling:

```
EVP_RAND *rand = EVP_RAND_fetch(NULL, "HMAC−DRBG", NULL);
EVP_RAND_CTX *rctx = EVP_RAND_CTX_new(rand);
```

EXAMPLES

```
EVP_RAND *rand;
EVP_RAND_CTX *rctx;
unsigned char bytes[100];
OSSL_PARAM params[3], *p = params;
unsigned int strength = 128;

rand = EVP_RAND_fetch(NULL, "HMAC−DRBG", NULL);
rctx = EVP_RAND_CTX_new(rand, NULL);
EVP_RAND_free(rand);

*p++ = OSSL_PARAM_construct_utf8_string(OSSL_DRBG_PARAM_MAC, SN_hmac, 0);
*p++ = OSSL_PARAM_construct_utf8_string(OSSL_DRBG_PARAM_DIGEST, SN_sha256, 0);
*p = OSSL_PARAM_construct_end();
EVP_RAND_instantiate(rctx, strength, 0, NULL, 0, params);

EVP_RAND_generate(rctx, bytes, sizeof(bytes), strength, 0, NULL, 0);

EVP_RAND_CTX_free(rctx);
```

CONFORMING TO

NIST SP 800−90A and SP 800−90B

SEE ALSO

EVP_RAND (3), ''PARAMETERS'' in **EVP_RAND** (3)

COPYRIGHT