NAME

EVP_aria_128_cbc, EVP_aria_192_cbc, EVP_aria_256_cbc, EVP_aria_128_cfb, EVP_aria_192_cfb, EVP_aria_256_cfb, EVP_aria_128_cfb1, EVP_aria_192_cfb1, EVP_aria_256_cfb1, EVP_aria_128_cfb8, EVP_aria_192_cfb8, EVP_aria_256_cfb8, EVP_aria_128_cfb128, EVP_aria_192_cfb128, EVP_aria_256_cfb128, EVP_aria_128_ctr, EVP_aria_192_ctr, EVP_aria_256_ctr, EVP_aria_128_ecb, EVP_aria_192_ecb, EVP_aria_256_ecb, EVP_aria_128_ofb, EVP_aria_192_ofb, EVP_aria_256_ofb, EVP_aria_128_ccm, EVP_aria_192_ccm, EVP_aria_256_ccm, EVP_aria_128_gcm, EVP_aria_192_gcm, EVP_aria_256_gcm, - EVP ARIA cipher

SYNOPSIS

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

const EVP_CIPHER *EVP_ciphername(void)
```

EVP_ciphername is used a placeholder for any of the described cipher functions, such as *EVP_aria_128_cbc*.

DESCRIPTION

The ARIA encryption algorithm for EVP.

EVP_aria_128_cbc(), **EVP_aria_192_cbc()**, **EVP_aria_256_cbc()**, **EVP_aria_128_cfb()**, **EVP_aria_192_cfb()**, **EVP_aria_256_cfb()**, **EVP_aria_128_cfb1()**, **EVP_aria_192_cfb1()**, **EVP_aria_256_cfb1()**, **EVP_aria_128_cfb8()**, **EVP_aria_192_cfb8()**, **EVP_aria_256_cfb8()**, **EVP_aria_128_cfb128()**, **EVP_aria_192_cfb128()**, **EVP_aria_256_cfb128()**, **EVP_aria_128_ctr()**, **EVP_aria_192_ctr()**, **EVP_aria_256_ctr()**, **EVP_aria_128_ecb()**, **EVP_aria_192_ecb()**, **EVP_aria_256_ecb()**, **EVP_aria_128_ofb()**, **EVP_aria_192_ofb()**, **EVP_aria_256_ofb()**

ARIA for 128, 192 and 256 bit keys in the following modes: CBC, CFB with 128-bit shift, CFB with 1-bit shift, CFB with 8-bit shift, CTR, ECB and OFB.

EVP_aria_128_ccm(), **EVP_aria_192_ccm()**, **EVP_aria_256_ccm()**, **EVP_aria_128_gcm()**, **EVP_aria_192_gcm()**, **EVP_aria_256_gcm()**,

ARIA for 128, 192 and 256 bit keys in CBC-MAC Mode (CCM) and Galois Counter Mode (GCM). These ciphers require additional control operations to function correctly, see the "AEAD Interface" in **EVP_EncryptInit**(3) section for details.

NOTES

Developers should be aware of the negative performance implications of calling these functions multiple times and should consider using **EVP_CIPHER_fetch**(3) instead. See "Performance" in **crypto**(7) for further information.

RETURN VALUES

These functions return an **EVP_CIPHER** structure that contains the implementation of the symmetric cipher. See **EVP_CIPHER_meth_new**(3) for details of the **EVP_CIPHER** structure.

SEE ALSO

evp(7), **EVP_EncryptInit**(3), **EVP_CIPHER_meth_new**(3)

COPYRIGHT