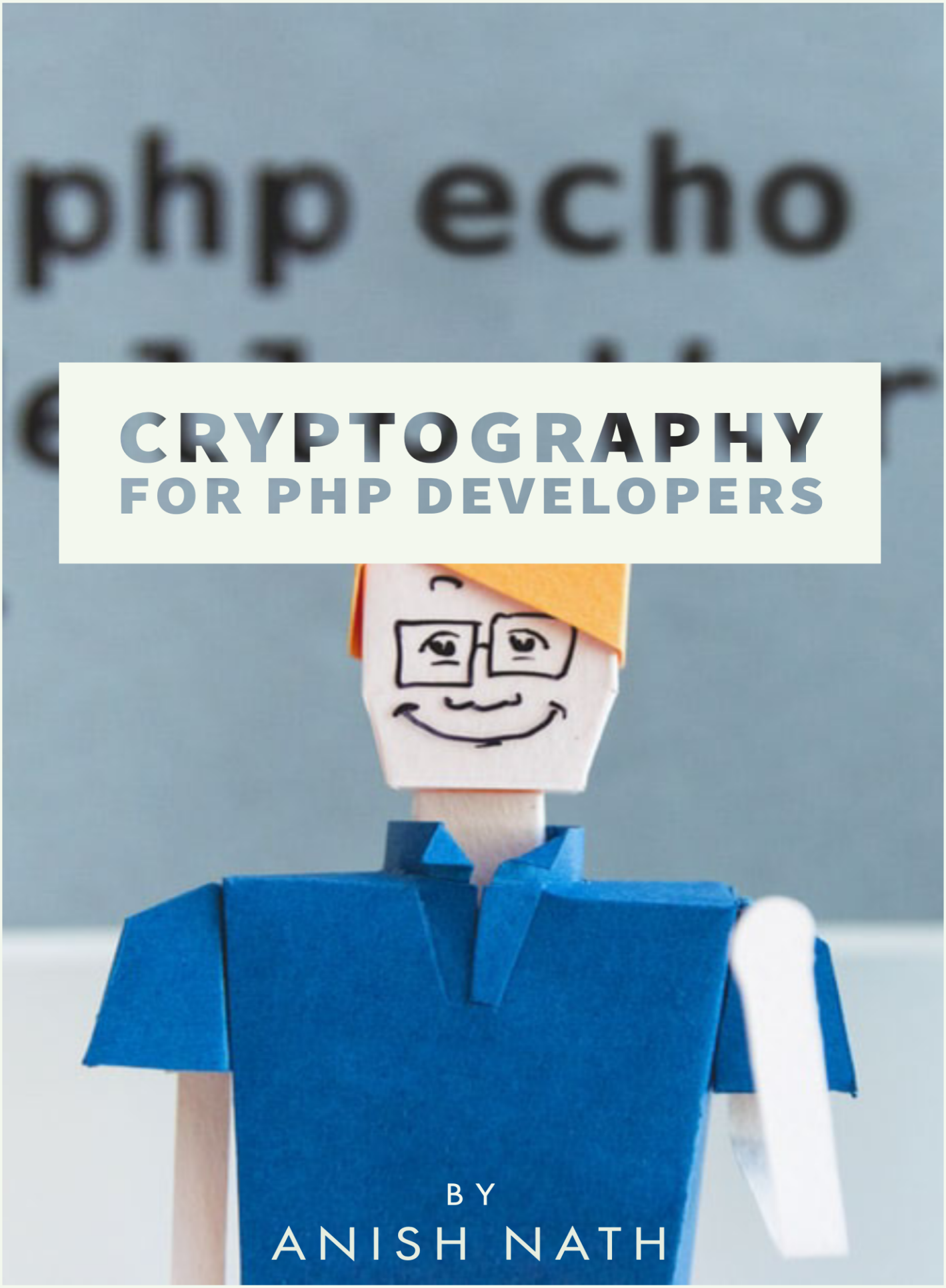

CRYPTOGRAPHY
FOR PHP DEVELOPERS

BY

ANISH NATH

Cryptography for PHP Developers

Anish Nath

This book is for sale at http://leanpub.com/cryptophp

This version was published on 2019-11-13

Contents

openssl_get_cipher_methods

openssl_get_cipher_methods: Gets available cipher methods.

- Supported PHP Versions (**PHP 5 >= 5.3.0, PHP 7**)

The Syntax

```
array  openssl_get_cipher_methods ([ bool  $aliases =  `FALSE` ] )
```

- **aliases**: Set to TRUE if cipher aliases should be included within the returned array.
- **Returns** An array[1] of available cipher methods.

Examples

openssl_get_cipher_methods() example shows how the available ciphers might look, and also which aliases might be available.

```
/**
 * Created by https://8gwifi.org
 * User: Anish Nath
 * Date: 2018-12-10
 * Time: 15:13
 */

<?php
$ciphers = openssl_get_cipher_methods();
$ciphers_and_aliases = openssl_get_cipher_methods(true);
$cipher_aliases = array_diff($ciphers_and_aliases, $ciphers);
print_r($ciphers);
print_r($cipher_aliases);
?>
```

The above example will output something similar to:

[1] http://php.net/manual/en/language.types.array.php

```
$ /usr/bin/php get_cipher_methods.php
/**
 * Created by https://8gwifi.org
 * User: Anish Nath
 * Date: 2018-12-10
 * Time: 15:13
 */
Array
(
    [0] => AES-128-CBC
    [1] => AES-128-CFB
    [2] => AES-128-CFB1
    [3] => AES-128-CFB8
    [4] => AES-128-CTR
    [5] => AES-128-ECB
    [6] => AES-128-OFB
    [7] => AES-128-XTS
    [8] => AES-192-CBC
    [9] => AES-192-CFB
    [10] => AES-192-CFB1
    [11] => AES-192-CFB8
    [12] => AES-192-CTR
    [13] => AES-192-ECB
    [14] => AES-192-OFB
    [15] => AES-256-CBC
    [16] => AES-256-CFB
    [17] => AES-256-CFB1
    [18] => AES-256-CFB8
    [19] => AES-256-CTR
    [20] => AES-256-ECB
    [21] => AES-256-OFB
    [22] => AES-256-XTS
    [23] => BF-CBC
    [24] => BF-CFB
    [25] => BF-ECB
    [26] => BF-OFB
    [27] => CAMELLIA-128-CBC
    [28] => CAMELLIA-128-CFB
    [29] => CAMELLIA-128-CFB1
    [30] => CAMELLIA-128-CFB8
    [31] => CAMELLIA-128-ECB
    [32] => CAMELLIA-128-OFB
    [33] => CAMELLIA-192-CBC
```

```
[34] => CAMELLIA-192-CFB
[35] => CAMELLIA-192-CFB1
[36] => CAMELLIA-192-CFB8
[37] => CAMELLIA-192-ECB
[38] => CAMELLIA-192-OFB
[39] => CAMELLIA-256-CBC
[40] => CAMELLIA-256-CFB
[41] => CAMELLIA-256-CFB1
[42] => CAMELLIA-256-CFB8
[43] => CAMELLIA-256-ECB
[44] => CAMELLIA-256-OFB
[45] => CAST5-CBC
[46] => CAST5-CFB
[47] => CAST5-ECB
[48] => CAST5-OFB
[49] => ChaCha
[50] => DES-CBC
[51] => DES-CFB
[52] => DES-CFB1
[53] => DES-CFB8
[54] => DES-ECB
[55] => DES-EDE
[56] => DES-EDE-CBC
[57] => DES-EDE-CFB
[58] => DES-EDE-OFB
[59] => DES-EDE3
[60] => DES-EDE3-CBC
[61] => DES-EDE3-CFB
[62] => DES-EDE3-CFB1
[63] => DES-EDE3-CFB8
[64] => DES-EDE3-OFB
[65] => DES-OFB
[66] => DESX-CBC
[67] => GOST 28147-89
[68] => RC2-40-CBC
[69] => RC2-64-CBC
[70] => RC2-CBC
[71] => RC2-CFB
[72] => RC2-ECB
[73] => RC2-OFB
[74] => RC4
[75] => RC4-40
[76] => RC4-HMAC-MD5
```

```
[77] => aes-128-cbc
[78] => aes-128-cfb
[79] => aes-128-cfb1
[80] => aes-128-cfb8
[81] => aes-128-ctr
[82] => aes-128-ecb
[83] => aes-128-gcm
[84] => aes-128-ofb
[85] => aes-128-xts
[86] => aes-192-cbc
[87] => aes-192-cfb
[88] => aes-192-cfb1
[89] => aes-192-cfb8
[90] => aes-192-ctr
[91] => aes-192-ecb
[92] => aes-192-gcm
[93] => aes-192-ofb
[94] => aes-256-cbc
[95] => aes-256-cfb
[96] => aes-256-cfb1
[97] => aes-256-cfb8
[98] => aes-256-ctr
[99] => aes-256-ecb
[100] => aes-256-gcm
[101] => aes-256-ofb
[102] => aes-256-xts
[103] => bf-cbc
[104] => bf-cfb
[105] => bf-ecb
[106] => bf-ofb
[107] => camellia-128-cbc
[108] => camellia-128-cfb
[109] => camellia-128-cfb1
[110] => camellia-128-cfb8
[111] => camellia-128-ecb
[112] => camellia-128-ofb
[113] => camellia-192-cbc
[114] => camellia-192-cfb
[115] => camellia-192-cfb1
[116] => camellia-192-cfb8
[117] => camellia-192-ecb
[118] => camellia-192-ofb
[119] => camellia-256-cbc
```

```
    [120] => camellia-256-cfb
    [121] => camellia-256-cfb1
    [122] => camellia-256-cfb8
    [123] => camellia-256-ecb
    [124] => camellia-256-ofb
    [125] => cast5-cbc
    [126] => cast5-cfb
    [127] => cast5-ecb
    [128] => cast5-ofb
    [129] => chacha
    [130] => des-cbc
    [131] => des-cfb
    [132] => des-cfb1
    [133] => des-cfb8
    [134] => des-ecb
    [135] => des-ede
    [136] => des-ede-cbc
    [137] => des-ede-cfb
    [138] => des-ede-ofb
    [139] => des-ede3
    [140] => des-ede3-cbc
    [141] => des-ede3-cfb
    [142] => des-ede3-cfb1
    [143] => des-ede3-cfb8
    [144] => des-ede3-ofb
    [145] => des-ofb
    [146] => desx-cbc
    [147] => gost89
    [148] => gost89-cnt
    [149] => gost89-ecb
    [150] => id-aes128-GCM
    [151] => id-aes192-GCM
    [152] => id-aes256-GCM
    [153] => rc2-40-cbc
    [154] => rc2-64-cbc
    [155] => rc2-cbc
    [156] => rc2-cfb
    [157] => rc2-ecb
    [158] => rc2-ofb
    [159] => rc4
    [160] => rc4-40
    [161] => rc4-hmac-md5
)
```

```
Array
(
    [23] => AES128
    [24] => AES192
    [25] => AES256
    [26] => BF
    [49] => CAMELLIA128
    [50] => CAMELLIA192
    [51] => CAMELLIA256
    [52] => CAST
    [53] => CAST-cbc
    [59] => DES
    [76] => DES3
    [77] => DESX
    [80] => RC2
    [116] => aes128
    [117] => aes192
    [118] => aes256
    [119] => bf
    [124] => blowfish
    [143] => camellia128
    [144] => camellia192
    [145] => camellia256
    [146] => cast
    [147] => cast-cbc
    [153] => des
    [170] => des3
    [171] => desx
    [179] => rc2
)
Process finished with exit code 0
```

In `openssl` You can get a list of available cipher methods by calling

```
$ openssl list-cipher-commands
```

The above example will output something similar to:

```
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
cast5-ofb
des
des-cbc
des-cfb
des-ecb
des-ede
des-ede-cbc
des-ede-cfb
des-ede-ofb
des-ede3
des-ede3-cbc
des-ede3-cfb
des-ede3-ofb
des-ofb
des3
desx
rc2
rc2-40-cbc
rc2-64-cbc
rc2-cbc
rc2-cfb
rc2-ecb
rc2-ofb
rc4
rc4-40
seed
```

openssl_get_md_methods

openssl_get_md_method: Gets available digest methods.

- Supported PHP Versions (**PHP 5 >= 5.3.0, PHP 7**)

The Syntax

```
array  openssl_get_md_methods ([ bool  $aliases =  `FALSE` ] )
```

- **aliases**: Set to TRUE if cipher aliases should be included within the returned array.
- **Returns** An array[2] of available digest methods.

Examples

openssl_get_md_methods() example Shows how the available digest might look, and also which aliases might be available.

```
<?php
/**
 * Created by https://8gwifi.org
 * User: Anish Nath
 * Date: 2018-12-12 * Time: 10:40
 * */
$digests = openssl_get_md_methods();
$digests_and_aliases = openssl_get_md_methods(true);
$digest_aliases = array_diff($digests_and_aliases, $digests);
print_r($digests);
print_r($digest_aliases);
?>
```

The above example will output something similar to:

[2] http://php.net/manual/en/language.types.array.php

```
$ /usr/bin/php get_md_methods.php
Array
(
    [0] => DSA
    [1] => DSA-SHA
    [2] => GOST 28147-89 MAC
    [3] => GOST R 34-11-2012 (512 bit)
    [4] => GOST R 34.11-2012 (256 bit)
    [5] => GOST R 34.11-94
    [6] => MD4
    [7] => MD5
    [8] => RIPEMD160
    [9] => SHA
    [10] => SHA1
    [11] => SHA224
    [12] => SHA256
    [13] => SHA384
    [14] => SHA512
    [15] => dsaEncryption
    [16] => dsaWithSHA
    [17] => ecdsa-with-SHA1
    [18] => gost-mac
    [19] => md4
    [20] => md5
    [21] => md_gost94
    [22] => ripemd160
    [23] => sha
    [24] => sha1
    [25] => sha224
    [26] => sha256
    [27] => sha384
    [28] => sha512
    [29] => streebog256
    [30] => streebog512
    [31] => whirlpool
)
Array
(
    [2] => DSA-SHA1
    [3] => DSA-SHA1-old
    [4] => DSS1
    [12] => RSA-MD4
    [13] => RSA-MD5
```

```
    [14] => RSA-RIPEMD160
    [15] => RSA-SHA
    [16] => RSA-SHA1
    [17] => RSA-SHA1-2
    [18] => RSA-SHA224
    [19] => RSA-SHA256
    [20] => RSA-SHA384
    [21] => RSA-SHA512
    [30] => dsaWithSHA1
    [31] => dss1
    [35] => md4WithRSAEncryption
    [37] => md5WithRSAEncryption
    [39] => ripemd
    [41] => ripemd160WithRSA
    [42] => rmd160
    [45] => sha1WithRSAEncryption
    [47] => sha224WithRSAEncryption
    [49] => sha256WithRSAEncryption
    [51] => sha384WithRSAEncryption
    [53] => sha512WithRSAEncryption
    [54] => shaWithRSAEncryption
    [55] => ssl2-md5
    [56] => ssl3-md5
    [57] => ssl3-sha1
)
```

In `openssl` You can get a list of available cipher methods by calling

```
$ openssl list-message-digest-commands
```

The above example will output something similar to:

```
md2
md4
md5
mdc2
rmd160
sha
sha1
```