# Author: Trevor Perrin

# See the LICENSE file for legal information regarding use of this file.

"""Abstract class for AES."""

class AES(object):

def \_\_init\_\_(self, key, mode, IV, implementation):

if len(key) not in (16, 24, 32):

raise AssertionError()

if mode != 2:

raise AssertionError()

if len(IV) != 16:

raise AssertionError()

self.isBlockCipher = True

self.block\_size = 16

self.implementation = implementation

if len(key)==16:

self.name = "aes128"

elif len(key)==24:

self.name = "aes192"

elif len(key)==32:

self.name = "aes256"

else:

raise AssertionError()

#CBC-Mode encryption, returns ciphertext

#WARNING: \*MAY\* modify the input as well

def encrypt(self, plaintext):

assert(len(plaintext) % 16 == 0)

#CBC-Mode decryption, returns plaintext

#WARNING: \*MAY\* modify the input as well

def decrypt(self, ciphertext):

assert(len(ciphertext) % 16 == 0)