

Forum:[Hack Forums](#)

Thread: [Rijndael Padding in Python](#)

000 - June 4, 2019, 8:13 am

I know where the problem is but I don't know how to fix it. The problem is with the padding. I have absolutely no idea about it and how it works. I tried searching online but nothing seemed to help.

Here's the actual code of my python:

```
Code from rijndael.cipher.crypt import new
from rijndael.cipher.blockcipher import MODE_CBC
import base64
```

```
PADDING = b'.'
```

```
def r_pad(payload, block_size=32):
    return payload + (block_size - len(payload) % block_size) * PADDING
```

```
KEY = 'lkiwrf897+22#bbtrm8814z5qq=498j5'
IV = '741952hheeyy66#cs!9hvjv887mxx7@8y'
plain_text = "A padded string to BLOCKSIZE length."
```

```
rjn = new(KEY, MODE_CBC, IV, blocksize=32)
encd = rjn.encrypt(r_pad(plain_text))
data = base64.b64encode(encd)
print(data)
```

```
rjn = new(KEY, MODE_CBC, IV, blocksize=32)
data = base64.b64decode(data)
decd = rjn.decrypt(r_pad(data))
print (decd)
```

This is the output:

Quote:

Dv0Y/AFXdFMIDrclDFCu8v5o9zAILNgyM+vO+PFeSrQWdzP1S1cumviFiEjNAjz5nJnMMC9lfxsBI71x5y+xCw==

A padded string to BLOCKSIZE length.....Å¿:è°?+n+«?Px?:æC+??Q+?«U_?â:~)ir

I need the output of the encrypted string to be something like this:

Quote:

Dv0Y/AFXdFMIDrclDFCu8v5o9zAILNgyM+vO+PFeSrpO8Ve82mdUcc4rkzp9afDYc75NmKsd4mdflt38kceOdA==

A padded string to BLOCKSIZE length

Can anyone please help? I am lost at this point. :/

HFDone - June 4, 2019, 11:06 am

<https://github.com/moeenz/rijndael>

you have usage there you dont need the r_pad function the blocksize is the length

```
KEY = 'lkirwf897+22#bbtrm8814z5qq=498j5'  
IV = '741952hheeyy66#cs!9hvjv887mxx7@8y'  
plain_text = "A padded string to BLOCKSIZE length."
```

```
rijn = new(KEY, MODE_CBC, IV, blocksize=32)  
encd = rjn.encrypt(plain_text)  
data = base64.b64encode(encd)  
print(data)
```

```
rijn = new(KEY, MODE_CBC, IV, blocksize=32)  
data = base64.b64decode(data)  
decd = rjn.decrypt(data)  
print(decd)
```

000 - June 4, 2019, 11:17 am

(Jun 4, 2019 11:06 AM)HFDone Wrote: [\[CITE\] https://github.com/moeenz/rijndael](https://github.com/moeenz/rijndael)
you have usage there you dont need the r_pad function the blocksize is the length

```
KEY = 'lkirwf897+22#bbtrm8814z5qq=498j5'  
IV = '741952hheeyy66#cs!9hvjv887mxx7@8y'  
plain_text = "A padded string to BLOCKSIZE length."
```

```
rijn = new(KEY, MODE_CBC, IV, blocksize=32)  
encd = rjn.encrypt(plain_text)  
data = base64.b64encode(encd)  
print(data)
```

```
rijn = new(KEY, MODE_CBC, IV, blocksize=32)  
data = base64.b64decode(data)  
decd = rjn.decrypt(data)  
print(decd)
```

I would need the string padded if the length of the string exceeds 32 characters, wouldn't I?

Here's what I came up with:

Possible Solution 1: Split the long string into multiple strings:

```
Code plain_text = b'A padded string to BLOCKSIZE length.'  
split_plain_text = plain_text.split()  
encd = "
```

```
rijn = new(KEY, MODE_CBC, IV, blocksize=32)
```

```
for x in split_plain_text:  
    encd = rjn.encrypt(x)  
    encd = base64.b64encode(encd)  
    print('Encrypted: ' + encd)
```

But that doesn't seem to work. I am getting this output:

Quote:Encrypted:

Encrypted:
Encrypted:
Encrypted:
Encrypted:
Encrypted:

* Another problem with this method would be if the string is something like "AStringWhichIsMoreThan32CharactersLongWithoutSpaces". It would be difficult to split it since I won't know the input.

Possible Solution 2: Maybe I should be converting the strings into bytes? Since some characters take more than 1 byte to represent? But this doesn't seem to be working as well:

Code `plain_text = b'A padded string to BLOCKSIZE length.'`
I don't think that's an actual way to convert string into bytes. But then again, I don't know how to. I have searched for it on Google but nothing seemed to help. Even tried `.encode()`.

HFDone - June 4, 2019, 11:52 am

i didnt get at first here how you do it
its not the same lib but you can get the idea
`pad = lambda s: s + (BLOCK_SIZE - len(s) % BLOCK_SIZE) * PADDING`
<http://www.codekoala.com/posts/aes-encryption-python-using-pycrypto>

000 - June 4, 2019, 11:57 am

(Jun 4, 2019 11:52 AM)HFDone Wrote: [\[CITE\]](#) Snipped
Thank you so much but I figured I was making a simple mistake. Here's what fixed my issue:

Code `PADDING = b'\x00'`

```
def pad(payload, block_size=32):  
    return payload + (block_size - len(payload) % block_size) * PADDING
```

```
def un_pad(payload):  
    return payload.replace(PADDING, "")
```

```
KEY = 'lkirwf897+22#bbtrm8814z5qq=498j5'  
IV = '741952hheeyy66#cs!9hvjv887mxx7@8y'  
plain_text = "A padded string to BLOCKSIZE length."
```

```
rjn = new(KEY, MODE_CBC, IV, blocksize=32)  
encd = rjn.encrypt(pad(plain_text))  
data = base64.b64encode(encd)  
print(data)
```

```
rjn = new(KEY, MODE_CBC, IV, blocksize=32)  
data = base64.b64decode(data)  
decd = rjn.decrypt(un_pad(data))  
print (decd)
```

It's not the proper way to do it, I know but at least it works.

A better approach would be:

```
Code def pad(payload, block_size=32):  
    pad_len = block_size - len(payload) % block_size  
    return payload + pad_len * chr(pad_len)
```

HFDone - June 4, 2019, 12:21 pm

dont use replace for unpadding as you can have the padding in your encrypt data if you dont use base64 or if you are using base64 char for padding
use rstrip as in the link so you dont have to deal with this problem

000 - June 4, 2019, 12:24 pm

(Jun 4, 2019 12:21 PM)HFDone Wrote: [\[CITE\]](#) dont use replace for unpadding as you can have the padding in your encrypt data if you dont use base64 or if you are using base64 char for padding
use rstrip as in the link so you dont have to deal with this problem

You're right. Thank you again. Have a nice day! :)