

Forum: [Hack Forums](#)

Thread: [RC4 Encryption implement](#)

M5DD - October 26, 2016, 10:20 am

<http://pastebin.com/ETF01RuV>

Goodies - October 26, 2016, 11:04 am

Wow that is ugly as shit. Looks like you're trying to do this with as little whitespace as possible. THAT's utterly unnecessary. Expand it, name your variables something meaningful instead of just letters, and use standard things like binascii's hexlify/unhexlify instead of some weird custom shit.

M5DD - October 26, 2016, 12:01 pm

(Oct 26, 2016 11:04 AM)Goodies Wrote: [\[CITE\]](#) Wow that is ugly as shit. Looks like you're trying to do this with as little whitespace as possible. THAT's utterly unnecessary. Expand it, name your variables something meaningful instead of just letters, and use standard things like binascii's hexlify/unhexlify instead of some weird custom shit.

I like my code to be non dependent on any module when it doesn't need to be. Here is the code rewritten to your criteria, enjoy.

<http://pastebin.com/nMwLx26B>

Goodies - October 26, 2016, 12:19 pm

(Oct 26, 2016 12:01 PM)M5DD Wrote: [\[CITE\]](#) I like my code to be non dependent on any module when it doesn't need to be. Here is the code rewritten to your criteria, enjoy.

<http://pastebin.com/XKn6vWa8>

That looks much better aesthetically.

Also, since binascii is apart of the standard library, it's not really a dependency...

Now, the gripe I have is that I'm not sure you're aware of how RC4 is supposed to function. It's a stream cipher meaning each time I use it, it should give me the next part of the stream. In Python, I'd probably implement it as a generator (or at least an object with a `a_next` function defined) that simple takes the key on initialization and returns the next byte.

Finally, it wouldn't work with Python3 which should really be your primary concern now.

M5DD - October 26, 2016, 12:25 pm

(Oct 26, 2016 12:19 PM)Goodies Wrote: [\[CITE\]](#) That looks much better aesthetically.

Also, since binascii is apart of the standard library, it's not really a dependency...

Now, the gripe I have is that I'm not sure you're aware of how RC4 is supposed to function. It's a stream cipher meaning each time I use it, it should give me the next part of the stream. In Python, I'd probably implement it as a generator (or at least an object with a `a_next` function defined) that simple takes the key on initialization and returns the next byte.

Finally, it wouldn't work with Python3 which should really be your primary concern now.

I'm learning cryptography, essentially, I converted pseudo code to python code, I'm still grasping the basics of ciphers, so please forgive me there. As for compatibility for python 3, which I'm not familiar with, I'll try to fix it, however I'll look at the python changelog and attempt a solution. Thanks for the

suggestions, I've been looking to improve my code.

Update: I managed to fix the decryption function for 3.x, however the encrypt function is still pretty funky, am working on it.

HFDone - October 26, 2016, 1:22 pm

rc4 should enc dec with the same function

M5DD - October 26, 2016, 1:42 pm

(Oct 26, 2016 01:22 PM)HFDone Wrote: [\[CITE\]](#) rc4 should enc dec with the same function
You're correct, however the conversion to hex and conversion back to decimal require two functions, I will alter accordingly, thanks. This is also compatible with python 3.x

<http://pastebin.com/ETF01RuV>

SilenceHacker - October 27, 2016, 3:46 am

I'm sorry for being a script kiddie but what does this do?

M5DD - October 27, 2016, 5:01 am

(Oct 27, 2016 03:46 AM)SilenceHacker Wrote: [\[CITE\]](#) I'm sorry for being a script kiddie but what does this do?

It allows you to encrypt / decrypt with the same function, it's rather useful for small programs when you need a simple encryption method without external modules. Try encrypting "test" with key "test", then decrypting the result with the same function and the same key.