

Re: Is this secure?

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2004-03/2463.html>

From: Jean-Luc Cooke (jlcooke_at_engsoc.org)

Date: 03/31/04

Date: 31 Mar 2004 14:43:14 GMT

John Burton <john.burton@jbmil.com> wrote:

- > *If I want to encrypt and transmit some data from a program running on*
- > *one computer to a remote one over the internet.*

Sounds like an assignment to me. :)

- > *1. Generate an RSA style public and private key pair (Using 1024 but*
- > *modulus in my case but that's variable anyway) and put the public key on*
- > *the machine I want to encrypt data on.*

Make sure $\phi(\phi(p * q))$ is large. This is a common recommendation to prevent Pollard $p-1$ attacks.

- > *2. Generate a 128 bit random number (I know this has to be done properly*
- > *and it's the point of my question)*
- > *3. Encrypt the data to be encrypted using AES with the 128 bit number as*
- > *they using CFB mode.*

Any reason why you chose this in stead of CBC or CTR? Why don't you add a MAC to the end of that? OMAC is nice.

- > *4. Turn the 128 bit random number into an integer and encrypt it using*
- > *the RSA algorithm and the public key.*

Make sure you use a good padding scheme (see PKCS#1.5 or whatever the acronym for the new one is)

- > *5. Send the encrypted key and encrypted data together as the encrypted*
- > *file. (In this base by concatenating them and base64 encoding)*

Sounds good... base64 is a "good thing" when using HTTP or other protocols. But if you're just making a file or a socket stream, you can just send the bytes raw and save 25% of the transmission overhead.

- > *I realise the general method is quite a standard way of encrypting*
- > *things, I just want to make sure I've not missed anything and in*
- > *particular that just sending the encrypted key and data in this form is*
- > **safe*.*

Re: Is this secure?

sci.crypt: Re: Is this secure?

Not really. You got most of it. Just add padding to your RSA operation, add a MAC to your data. Optionally you can get ride of base64 (depends on application) and make sure your modulus is strong against $p-1$ attacks (which 999 times out of 1000 it is).

*> I'm looking for this to be 'secure' as a general method so that if one
> message were cracked it wouldn't help with another, and I'm looking for
> it to be generally pretty secure against attacks taking place now rather
> than theoretical attacks taking place in the future.*

*> It's partly something I'm doing as a learning excercise as well as
> hopefully being something useful for myself in future so while I'd
> welcome any comments about products or other methods to use I'm
> particularly interested in what might be wrong if anything with this, or
> any information about where to look further.*

Well written post. Hope you get your answer.

JLC

--