

## Re: Another Dumb Idea for Debunking...

---

*Source:* <http://www.derkeiler.com/Newsgroups/sci.crypt/2006-10/msg00983.html>

---

- *From:* "Bill Cox" <[bill@xxxxxxxxxxxxxx](mailto:bill@xxxxxxxxxxxxxx)>
  - *Date:* 31 Oct 2006 03:00:55 -0800
- 

Sebastian Gottschalk wrote:

Bill Cox wrote:

You said RC4 could be used securely if you are careful. Can you elaborate, since I'm going to try to use it in TinyCrypt?

Dude, where's your Google gone?

Anyway, RC4 has some little disadvantages:

- It's a stream cipher. You have to make sure to never reuse a stream. This also involves using an IV.
- It has weak keys, so only use the 128 bit variant and discard the first 256 bytes.
- It has statistical weaknesses, so don't do too long without rekeying.

I've switch TinyCrypt to using it. It's more complicated than what I had, but it's faster, and better analyzed. By applying it twice, it's no longer a stream cipher. The first encryption is with a random 256-byte key, which is pre-pended to the data for the second pass. This doubles the run-time, but hopefully gets around the issue of using unique keys each time. Also, the first 256 bytes are just the encrypted random key, which doesn't correlate to the user's data.\

WTF is lzop? What about using Zlib?

Check out <http://www.oberhumer.com/opensource/lzo>. Basically, it's lower compression, but 4-5X faster than zlib. Also, it's been around a long time, and is included in most Linux distros.

Regardless of the lower-level encryption, TinyCrypt hopefully adds security by randomizing the input data by a first encryption with a

Re: Another Dumb Idea for Debunking...

random key, and then encrypting with the user's actual key. Is this good enough, or are there other steps I should take?

This is bad enough, and you should take fewer steps.

Sounds good, but how can I take fewer steps and still feel reasonably OK about having a ton of TinyCrypt encrypted files using the same key on my hard-disk? And, is this enough to feel OK about it?

Thanks for the feedback -- Bill

.