

Re: My little something...

Re: My little something...

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2006-07/msg00620.html>

- *From:* "Tom St Denis" <tomstdenis@xxxxxxxxxx>
 - *Date:* 17 Jul 2006 09:53:48 -0700
-

Markus Jansson wrote:

Tom St Denis wrote:

Where did you get a 1024 bit curve?

From my head. :)

Where it should remain.

And why are you using it with
256-bit ciphers?

For security margin, since ECC is kinda new field. Currently 512bit ECC would give same complexity as 256bit symmetric for bruteforcing I recall.

ECC is an older field than Twofish or AES designs are [AES is based off a 1995 Ph.D. dissertation]. Your logic escapes me.

Even suppose there is an attack on ECC, why do you assume it will be a sqrt/sqrt attack? If it goes sub-exp you could see 1024-bit ECC being much weaker than 256-bits.

– Twofish (LRW) & Serpent (CTR) as symmetric cipher.

You mean you offer two ciphers in different modes or that you chain it?

Two ciphers in different modes with different, independent keys.

Re: My little something...

Re: My little something...

And this is supposed to be more secure or just harder to implement correctly?

– Whirlpool as hash function (if 256bits is needed, output of Whirlpool is divided into two parts which are XOR:ed together).

Truncation is better than that.

Why? If there is "not-randomness" in the output, you might end up with the piece that is "not-so-random". When you take the whole package, split it half and XOR it, its unlikely that output would be

"less-random" than the original package.

Why do you assume that breaks will be so predictable. What if the break is that there is a correlation between halves of the output? Then the XOR would no more entropy than just one of the halves.

If Whirlpool is a secure hash than truncation **MUST BE** just as secure as XOR'ing the halves. Otherwise all bets are off.

Also do you plan on encrypting 2^{256} files per password?

?

You don't need a 512-bit salt unless you plan to use your password 2^{256} times.

PRNG:s used

– Yarrow & Mersenne Twister & Fortuna (with Twofish) & Blum-Blum-Shub & RtlGenRandom and user interactiveTM (mouse movement).

WHY!!!!

To get random numbers for ciphers. I dont want to end up in situation where only one PRNG is used and then after few years we know it has actually been broken or is otherwise bad PRNG. Combine few different PRNG:s and failure in one or two does not compromise security.

Re: My little something...

Re: My little something...

This is nonsense. If anything your seeding data will be the source of the compromise not the PRNG. Using multiple PRNGs from the same bad entropy source is not going to make it more secure.

What are your specific goals you are trying to accomplish with cryptography?

Kill some time and test my brains a bit. :p

You really need to learn about the primitives, their goals and limitations before you even think about putting them together.

But with this one, as I said, I was thinking of upgrading current (lets say PGP) encryption to littlebit more paranoid mode, just thinking did I get most of it right or wrong...

If you want to upgrade PGP start with existing protocols. Integrate GF(p) ECC into it and start using the Whirlpool and SHA-2 series hashes.

The goal of a cryptographer is to engineer [e.g. analyze, design, develop] a cryptosystem that meets the needs of the user while not blowing budgets.

Using "big keys and lots of bits" doesn't mean your more secure. A bank with ECC-1024 and AES-512 with plywood walls on the vault isn't all of a sudden "secure".

What if I told you for the next decade you can easily protect secrets with an 80-bit symmetric key? Would that blow your mind?

What if I told you that essentially, barring advancements in the practicality of QC, 128-bit keys will still be useful for your great grand childrens secrets?

Using the "big number theorem" is the first sign of a person who just doesn't get it. You're looking at it from a "what crypto can I throw around" instead of a "what am I trying to accomplish" point of view.

Tom

.

Re: My little something...