

Re: Hardware True Random Number Generator design / concept

Source: <http://sci.tech-archive.net/Archive/sci.electronics.design/2005-06/msg03088.html>

- *From:* The Real Andy <will_get_back_to_you_on_This>
 - *Date:* Thu, 23 Jun 2005 14:32:35 +1000
-

On Tue, 21 Jun 2005 16:42:07 -0700, Yoy G0 <yoyg0@xxxxxxxxxxxx> wrote:

>-----BEGIN PGP SIGNED MESSAGE-----
>Hash: SHA1
>
>On Wed, 22 Jun 2005 06:49:45 +1000, you wrote:
>[snip]
>> Have you considered using a Psuedo RNG with hardware entropy for
>> seeding? There is plenty of great information out there to do this,
>> and it saves on having to buy hardware. Do a google for Mersenne
>> Twister, very good algorithm, long cycle.
>>
>> By the way, a good statistical package for testing is R. Its free
>> and its very powerful.
>
>I was thinking about a possibility of using that method
>for producing key material for One Time Pad (OTP – cryptography),
>but Matt Mahoney ommented in sci.crypt so
>(POSTING – Re: Secure Data & Communication Project):
>
>That is not one time pad. Not that it can't be done securely,
>but you don't have the theoretical secrecy against an attacker
>with unlimited computing power that OTP offers.
>With unlimited power the attacker can try all possible seeds
>(since there are only a finite number of them)
>and find the one that decrypts to something sensible.
>All the wrong decryptions will look like random data.
>With OTP all plaintexts are equally likely,
>including all the sensible ones, so there is no way to
>tell which one is correct.
>OTP will require a hardware random number
>generator for every bit of the keystream.

Given any number of unlimited resources, one can crack any cryptographic system. You need to dertermine your requirements and then make a decision based on how much money you want to spend and how much development time you wish to put in and how secure you require the system to be.

Re: Hardware True Random Number Generator design / concept

>
>Also, I found the following in Wikipedia:
>
>http://en.wikipedia.org/wiki/Mersenne_twister
>
>"Unlike Blum Blum Shub, the algorithm in its native form
>is not suitable for cryptography. For many other
>applications, however, it is fast becoming the
>random number generator of choice."
>
>I don't know why Mersenne Twister is
>not suitable for cryptography.
>Any ideas?
>

The reason they state that MT is not cryptographically secure is because it is a linear RNG. This means after a finite amount of time the sequence will be restarted and can become predictable.

A secure hashing algorithm can be used to circumvent this, but as with any PRNG, there will always be a finite cycle. >

See <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/efaq.html> for more detail. I have used MT many times, and usually randomly throw away numbers so that the sequence is less predictable.

• *Follow-Ups:*

- ◆ **[Re: Hardware True Random Number Generator design / concept](#)**
 ◇ From: Guy Macon
- ◆ **[Re: Hardware True Random Number Generator design / concept](#)**
 ◇ From: Robert Baer

• *References:*

- ◆ **[Re: Hardware True Random Number Generator design / concept](#)**
 ◇ From: The Real Andy
- Prev by Date: **[Re: Butterworth Filter](#)**
- Next by Date: **[Re: Bizzare behaviour from SG/UC3525](#)**
- Previous by thread: **[Re: Hardware True Random Number Generator design / concept](#)**

Re: Hardware True Random Number Generator design / concept

- Next by thread: ***Re: Hardware True Random Number Generator design / concept***
- Index(es):
 - ◆ ***Date***
 - ◆ ***Thread***