C3C Gavin Delphia

LtCol Werner

Math 340, T4

20 Mar 2012

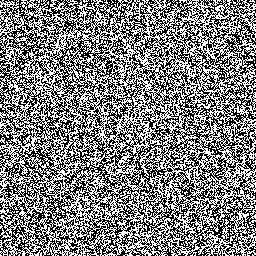
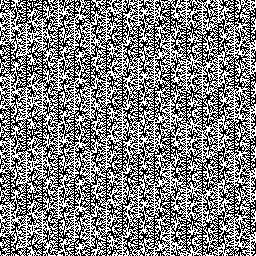
Pseudorandom and Random Numbers

In computer science and more specifically cryptology, random numbers are critical in creating secure methods of encryption. One of the major challenges cryptologists face in generating random numbers is how to determine if a sequence of numbers is truly random. It may be tempting to think that a computer is a random number generator (RNG) but in reality the numbers generated by computers are actually pseudo random. Since a computer is a state machine, it follows that everything a computer does has a set path that determined how it got there and where it will go next. This means that a computer can’t ever have something random happen like the generation of a random number. Even with pseudo random numbers cryptologists can generate secure encryption methods as long as they thoroughly test the generation method for weaknesses. However, even the most secure methods of generation will still produce a sequence of numbers with properties that truly random sequences should not have.

Since randomness is critical in the creation of secure encryption methods, it is necessary to be able to determine when a sequence of numbers is truly random. According to the National Institute of Science and Technology “a random number sequence is one in which all possible numbers have the same probability of occurring and that they occur independently of other numbers in the sequence.” (NIST) Occurring independently means that no matter how many numbers of the sequence are known it should be impossible to predict any other numbers in the sequence. Truly random numbers are generated by sources such as instruments that measure atmospheric data or in earlier times a camera that analyzed the formations of bubbles in a lava lamp. Since the atmospheric data and the shape of bubbles cannot be accurately predicted or duplicated they fit the classification of truly random number sources. Pseudo random number generators (PRNG) on the other hand can be predicted and duplicated.

Being able to predict a random number sequence or even duplicate that sequence of numbers is a problem for cryptologists. Since pseudo random numbers are generated using initial inputs called seeds which are then processed through complex algorithms, it is possible to predict and re-generate these sequences of random numbers. According to NIST, one downside to PRNGs is “the seed itself must be random and unpredictable…a PRNG should obtain its seed from the outputs of a RNG” (NIST). Since the algorithm used to generate these numbers is likely to be known, the seed must itself be random or the security of the sequence would be compromised. This reliance on the initial seed also means that the outputs of a PRNG must not be able to determine the initial seed used to generate the sequence. Another downside to PRNGs is that there are only so many cycles an algorithm can experience before it eventually repeats in some way.

Repeating in pseudo random numbers has real world implications especially in the field of RSA encryption. According to Burt Kaliski, chief scientist at RSA laboratories “RSA encryption is done with pairs of different keys, a public encryption key and a private decryption key” (Kaliski). The idea is that anyone can encrypt data but only the intended recipient with the paired private key can then read the data. This is how many online data transactions are kept secure from unintended recipients or hackers. A recent study of RSA public key generation determined that “Of 6.4 million distinct RSA moduli, more than 1.1% occur more than once, some of them thousands of times” (Lenstra) Since RSA key pair generation involves multiplying two large random primes together to get a moduli, if two keys end up with the same moduli it means they were created with the same prime numbers. Lenstra also points out those moduli that share even a single factor become insecure because together they can be used to quickly determine the unknown factor. The flaw in the PRNG that creates the large random primes for the RSA keys could allow someone to compromise the data security of someone else if they discovered they shared the same moduli or a factor. Since the moduli repeat sometimes more than 1000 times it is possible that the flaw with the PRNG is with its seed conditions not being truly random or that the probability of the algorithm cycle repeating is too large for the number of keys that have been generated. The website random.org demonstrated an example of a PRNG that repeats too frequently by generating an image based off the PRNGs output. The image below on the right is from a PRNG while the image on the left is from an atmospheric RNG.



The image on the right visually demonstrates the repetition and resulting pattern of the PRNG while the image on the left from the RNG does not have a repeating pattern. These pictures demonstrate the hazards of using a PRNG over a RNG and show how serious the current RSA situation is just on a less frequent scale of repetition.

Both PRNGs and RNGs have their place in the field of cryptology and it is important to be able to recognize the differences between the sequences of numbers from each generator. True randomness can only exist when the sequence cannot be predicted or intentionally duplicated, while pseudo random sequences can be predicted and duplicated using algorithms along with seeds. This predictability and periodicity that PRNGs have are their greatest weakness in terms of cryptographic security. It is also what makes a sequence of pseudo random numbers unique from truly random numbers.

Works Cited

“Introduction to Randomness.” *Random.*  Random.org. n.d.  Web.  15 Mar. 2012.

Kaliski, Burt. *The Mathematics of the RSA Public-Key Cryptosystem*. *Math Awareness*. RSA Laboratories. Web. 19 Mar. 2012. <http://mathaware.org/mam/06/Kaliski.pdf>.

Lenstra, Arjen K., James P. Hughes, Maxime Augier, Joppe W. Boss, Thorsten Kleinjung, and Christophe Wachter. *Ron Was Wrong, Whit Is Right*. *Cryptology EPrint Archive*. Web. 19 Mar. 2012. <http://eprint.iacr.org/2012/064.pdf>.

Rosen, Kenneth H.  *Discrete Mathematics and Its Applications*. 6 ed.  New York: McGraw Hill, 2007.  Print.

Rukhin, Andrew L.  “A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.”  Washington: U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, 2001.  Print.

Science Clarified. "Renewable Energy Is Beneficial for the Environment." *Opposing Viewpoints: Renewable Energy*. Ed. Jacqueline Langwith. Detroit: Greenhaven Press, 2009. *Opposing Viewpoints Resource Center*. Gale. USAF Academy Library. 14 Mar. 2011 <http://find.galegroup.com/>.

Documentation

I shared sources with the cadets that share my topic, C3C Christopher McBride, C3C Anthony Canino and C3C Samuel Kiekhaefer. I talked with LtCol Werner and he gave me the idea of talking about the RSA flaw that was recently discovered. I copied the citation of Introduction to Randomness, Rosen, and Rukhin from C3C Christopher McBride as he emailed the citations with his sources. I did not receive any other help on this assignment.