Random Number Generator

Andy Chen

University of California, Santa Barbara

Introduction

- Does true randomness exist scientifically?
- If you know all the elements about a dice: the table it will hit, the air around it, etc.. You will be able to correctly guess the outcome
- However, this is as close to "randomness" as we can get. The phenomenon of "true randomness in nature."

Introduction

- Requirements of Randomness:
- All values has equal probability of getting chosen next
- Result is independent from predecessors and successors

Motivation

- Lottery/betting and other games that require randomness needs it to set a fair environment for the players
- Science experiments and other tests require random data set
- Most importantly, randomization is used to ensure maximum security in cryptography

Security

- A truly random key is one where each key has no ties on how it is generated
- The only way for an attacker to obtain such key is brute force - forcefully attempt every single possible combination
- This on average takes 2^{m-1} for an m-bit code. This can take a very long time

Security

- Requirements for Security:
- R1 random numbers are statistcally independent and unbiased
  - All values in the sets has an equal and random chance of showing up
- R2 - all numbers are unpredictable. Meaning backward and forward security
- R3 - backward security
- R4 - forwards security

Random Number Generator

- Determnistic RNG
  - Also known as pseudorandom number generators
  - True random numbers can not be computed on deterministic machines

- True RNG
  - Randomness that comes from the "true randomness in nature."
  - Utilizes tools and equipments that can capture data from more than just a deterministic computer machine

Deterministic RNG

- Machines with no access to outside data can only run deterministic RNGs
- Defined as having the appearance of randomness, but nevertheless having a repeatable pattern
- The "pseudo" in pseudo implies that it only mimicks randomness
- DRNG are computed using algorithms: Block Cipher, Linear congruential generators, Linear and nonlinear feedback shift registers, Number-theoretic RNGs, etc.

DNRG Algorithm: Middle Squares Method

- One of many DNRG algorithm is called the Middle Squares Method
- DNRG algorithm requires an initial seed, s_0, which can be initially obtained by acquiring something random: time in milliseconds, speed of keystrokes, or etc.
- Multiply the seed by itself, and take the middle result

DNRG Algorithm: Middle Squares Method

- 121 * 121 = 14641 midRes = 464 res = 464
- Next take the new middle result and repeat. Then append the new middle number onto the result.
- 464 * 464 = 215296 midRes = 529 res = 464529
- 529 * 529 = 279841 midRes = 984 res = 464529984
- repeat...

Deterministic RNG

- Advantages
  - Low memory. Only needs seed and algorithm. This forfeits the need to store a huge amount of random numbers
  - Hardware is not required like the case for true random number generators
  - Identical seed values will yeild to same output. This is good for some practices such as steam ciphers

Deterministic RNG

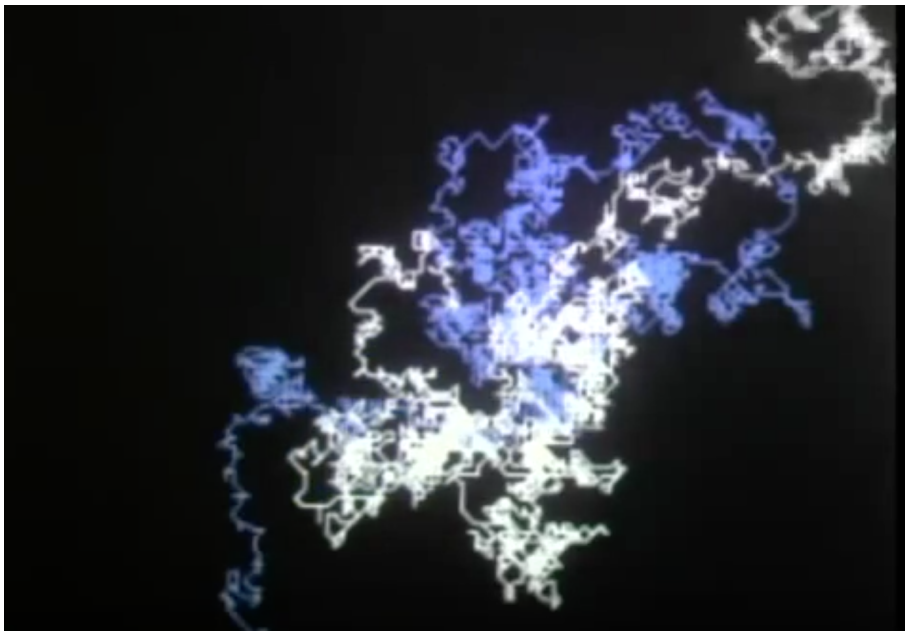
- Disadvantages:
  - Output is completely determined by the seed
  - Never really "completely" random
  - Pattern WILL repeat. Once the same seed comes back, the outputs will cycle over again.

Real RNG

- TNRG extract randomness from physical phenomenom and introduce it into a computer
- This randomness must rely on external hardware
  - Few computers have access to this kind of hardware
  - Some examples of such extrenal hardwares are sensors or noise detector
- There are some ways to mimick this effect without external hardware
  - Activities that are happening internally on the computer:
  - Time in milliseconds
  - How fast the user types

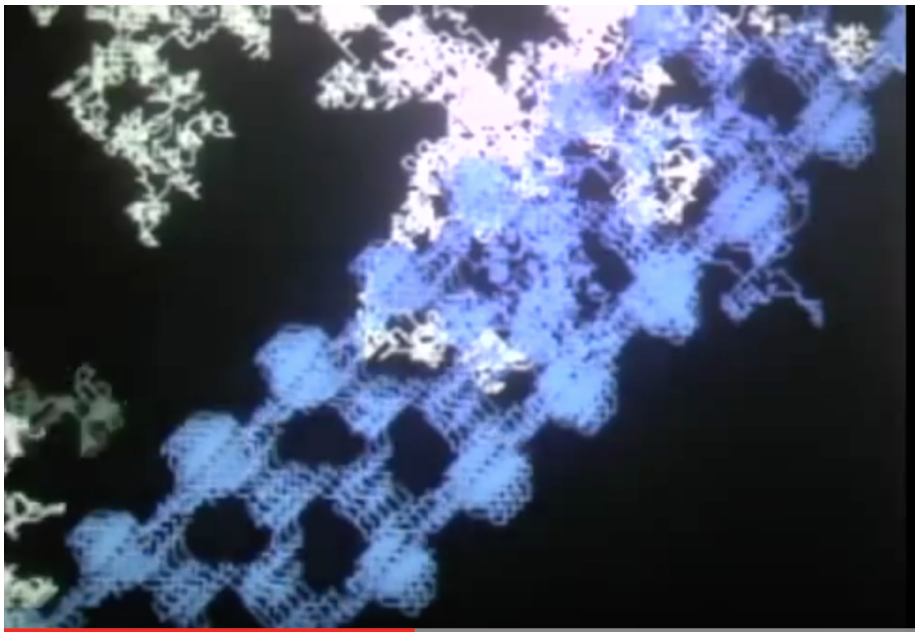
Testing

- Conducting small test to see the visual difference between DNRG and TNRG
- Known as "random walk," the lines moves in different direction based on the random number
- White: TNRG Blue: DRNG



Testing

- Previously picture looks fine. However, once you go on long enough you will find out DRNG will repeat
- Once the output of the algorithm is equal to the initial seed, it will repeat as shown below



Applications

Application	RNG To Use
Lottery and Gambling	Real RNG
Games	Real RNG
Random Sampling	Real RNG
Simulation and Models	Deterministic RNG
Security	Real RNG

Conclusion

- Random numbers are important as they are the basis for many security protocols
- Computers without external hardware can only generate pseudorandom numbers using various algorithms
- Computer applications are increasingly turning towards using physical data for getting truly random numbers