Pseudo-Random Number Generators

Sophia Lederer

Table of contents

01

Fundamentals of PRNGs

02

Common Types of PRNGs

03

Cryptographically Secure PRNGs

04

Evaluating Pseudorandom Sequences

05

Comparison: PRNG vs CSPRNG

06

Applications of PRNGs

Fundamentals of PRNGs

-A random number generator (RNG) is an algorithm or device designed to produce a sequence of numbers that exhibit characteristics of randomness

-True randomness is hard to achieve

-RNGs typically generate pseudo-random numbers

Classes of PRNGs: Hardware vs. Software

-Hardware-based: physical phenomena

-Software-based: mathematical algorithms

Common Types of PRNGs

Linear Congruential Generators

One of the simplest algorithms in this domain, generating pseudo-random integers through linear recurrence equations.

Middle Square Method

Involves squaring a number and extracting middle digits for subsequent numbers.

Mersenne Twister

Stands out for its extensive period and high randomness quality, yet its predictability once several outputs are known renders it unsuitable for cryptographic usage.

Common Types of PRNGs

Middle Square Method

Involves squaring a number and extracting middle digits for subsequent numbers.

Example:

675248

seed

Common Types of PRNGs

Middle Square Method

Involves squaring a number and extracting middle digits for subsequent numbers.

Example:

455959861504

seed^2

Common Types of PRNGs

Middle Square Method

Involves squaring a number and extracting middle digits for subsequent numbers.

Example:

455 504

959861

output

Common Types of PRNGs

Linear Congruential Generators

One of the simplest algorithms in this domain, generating pseudo-random integers through linear recurrence equations.

Middle Square Method

Involves squaring a number and extracting middle digits for subsequent numbers.

Mersenne Twister

Stands out for its extensive period and high randomness quality, yet its predictability once several outputs are known renders it unsuitable for cryptographic usage.

Common Types of PRNGs

Mersenne Twister

Stands out for its extensive period and high randomness quality, yet its predictability once several outputs are known renders it unsuitable for cryptographic usage.

Common Types of PRNGs

Linear Congruential Generators

One of the simplest algorithms in this domain, generating pseudo-random integers through linear recurrence equations.

Middle Square Method

Involves squaring a number and extracting middle digits for subsequent numbers.

Mersenne Twister

Stands out for its extensive period and high randomness quality, yet its predictability once several outputs are known renders it unsuitable for cryptographic usage.

Common Types of PRNGs

Linear Congruential Generators

One of the simplest algorithms in this domain, generating pseudo-random integers through linear recurrence equations.

$$X_{(n+1)} = (ax_n + c) \ \mathbf{mod} \ m$$

Importance of Prime Numbers in LCGs

-Influence period length and randomness quality

-Helps avoid undesirable congruencies

Significance of 2^{32} in LCGs

2^{32} corresponds to the maximum period length achievable with a 32-bit integer representation

(**2^{32}** - 1) = the largest number that can fit within 32 bits

2^{32} distinct pseudo-random numbers before repeating

Evaluating Pseudorandom Sequences

Chi-Square

A chi-square test is a statistical test used to determine whether there is a significant association between categorical variables by comparing observed frequencies with expected frequencies.

Autocorrelation

An autocorrelation test is a statistical method used to assess the degree of correlation between a sequence of values and a delayed version of itself, aiming to detect patterns or dependencies within the data.

Spectral Tests

A spectral test is a statistical analysis technique used to examine the presence of periodic patterns or cyclic components within a dataset by analyzing its frequency domain representation through methods like Fourier analysis.

Cryptographically Secure PRNGs

Fortuna

Fortuna is a cryptographically secure pseudorandom number generator (CSPRNG) designed to withstand various cryptographic attacks, employing a reseeding mechanism based on entropy sources to enhance randomness.

CryptGenRandom

CryptGenRandom is a cryptographic API function in Windows operating systems used to generate cryptographically secure pseudo-random numbers for various cryptographic purposes, ensuring high-quality randomness.

Yara

Yarrow is a cryptographically secure pseudorandom number generator (CSPRNG) designed to provide high-quality randomness for cryptographic applications by continuously updating its internal state based on environmental noise sources.

Traditional PRNGs vs CSPRNGs

-Traditional PRNGs lack cryptographic strength required for secure communication and data protection

-Cryptographically secure PRNGs are designed to withstand cryptographic attacks

Application of PRNGs

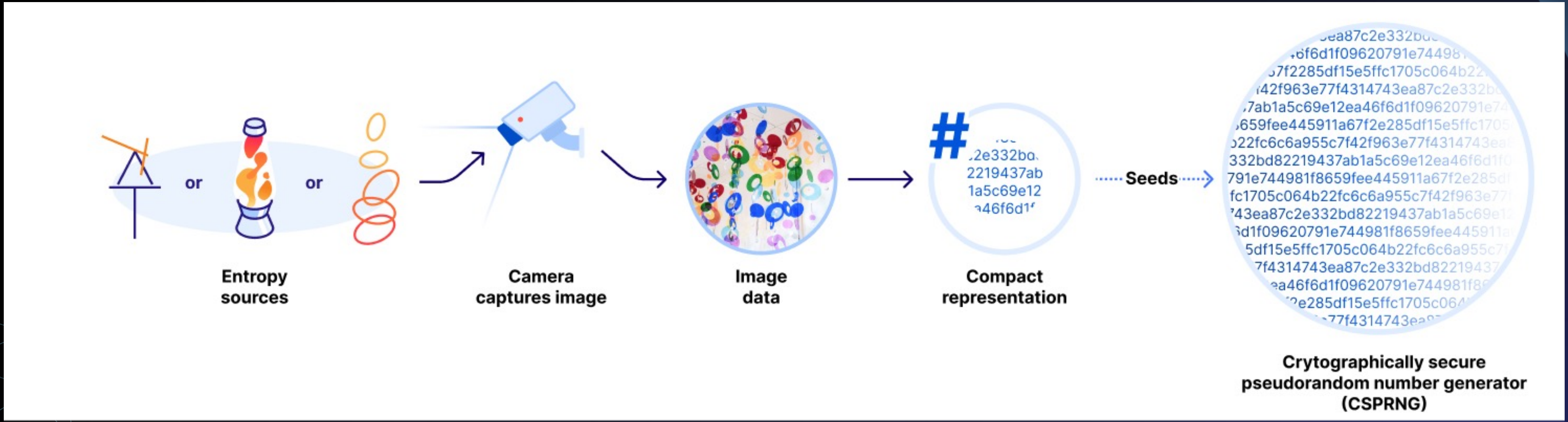
-Gaming

-Cryptographic Keys

-Secure communication

Cloudflare

How does it work?



Entropy sources

Camera captures image

Image data

Compact representation

Crytographically secure pseudorandom number generator (CSPRNG)

Conclusion

Resources

Cloudflare. (2024, March 30). Harnessing office chaos [Blog post]. Retrieved from https://blog.cloudflare.com/harnessing-office-chaos

Craddock, J. M., & Farmer, S. A. (1971). Two Robust Methods of Random Number Generation. *Journal of the Royal Statistical Society. Series D (The Statistician)*, *20*(3), 55–66. https://doi.org/10.2307/2986798

Juels, A., & Paar, C. (2007). The Blum Blum Shub pseudorandom generator. Dr. Dobb's Journal, 32(10), 108-109.

Knuth, D. E. (1997). The art of computer programming: Semi numerical algorithms (Vol. 2). Boston: Addison-Wesley.

L'Ecuyer, P., & Simard, R. (2007). TestU01: A C library for empirical testing of random number generators. ACM Transactions on Mathematical Software (TOMS), 33(4), 22.

Matsumoto, M., & Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equi-distributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation, 8(1), 3–30. https://doi.org/10.1145/272991.272995

Resources

Park, S. K., & Miller, K. W. (1988). Random number generators: Good ones are hard to find. Communications of the ACM, 31(10), 1192-1201. doi: 10.1145/63039.63042

Rosen, K. H. (2007). Discrete mathematics and its applications (7th ed.). New York, NY: McGraw-Hill.
Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., & Vo, S. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST special publication, 800-22rev1a. Retrieved from https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final

Sowey, E. R. (1972). A Chronological and Classified Bibliography on Random Number Generation and Testing. *International Statistical Review / Revue Internationale de Statistique*, *40*(3), 355–371. http://www.jstor.org/stable/1402472

Thanks!

Do you have any questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**