Deterministic Random Bit Generators NIST techniques comparison

Tomas Bedoya
Daniel Andres Jaimes Cardenas
Eder Leandro Carbonero Baquero
Universidad de los Andes, Colombia

***Abstract*—The following paper centers around several methods of random number generation. It aims to comprehend the essential concepts behind RNG, including pseudo and true random generation. Additionally, it explores different approaches to randomness that are used today that follow the NIST SP 800-90 standards, both reliant on entropy sources as well as algorithmic techniques, and compares their randomness with a series of statistical tests to determine which operate better and suggest appropriate uses for the different methods.**

General defintions

With the aim of providing a broader understanding of the various concepts addressed in this document, the following are definitions of some of the acronyms that will be mentioned throughout the text.

Acronym	Meaning
CTR_DRBG	Counter mode Deterministic Random Bit Generator
NIST	National Institute of Standards and Technology
AES	Advanced Encryption Standard
RBG	Random Bit Generator
DRBG	Deterministic Random Bit Generator
RNG	Random Number Generator
PRNG	Pseudorandom Number Generator
TRNG	True Random Number Generator
DRNG	Digital Random Number Generator
ENRNG	Enhanced Random Number Generator
HSM	Hardware Security Module

Table I
Acronyms definitions.

I. Introduction

Due to the deterministic nature of traditional computers, which is what most people and appliances use to this day and will continue being so for the foreseeable future, Random Number Generation (RNG) has proven to be a challenge. Due to the reliance of cryptographic algorithms, communication protocols and other security systems in random values, there's a constant race to find ever-better sources of randomness that follow the international standards of quality; the higher the randomness of a source, the greater the security it provides. This proves to be a challenge, however, for true randomness is difficult to find. Computers themselves cannot produce it and rely on Pseudo Random Number Generation (PRNG) to simulate random values. Other techniques, such as finding random measurements on certain like the time between user key strokes or mouse movements have proven to be unsatisfactory when put under statistical scrutiny (Mechalas, 2018). To guarantee that sensible processes utilize proper random sources, the NIST structured the SP 800-90 Standards for Random Number Generation (RNG) for safe RNG in cryptographic contexts. The following sections delve deep into several RNG methods, both of pseudo and true randomness, that follow these standards. They explore the different entropy sources they use (or don't), analyses how they extract random number chains from them, and evaluate the quality of their randomness through a series of statistical evaluations. Our aim is to understand better where RNG stands today, to evaluate how they fare against each other, and determine which of these methods work better for which applications.

II. Key Concepts

Most RNG methods can be classified into one of two categories. The first of these is Pseudo Random Number Generation, sometimes called Deterministic Random Bit Generation (DRBG) (Cao et al., 2022). A PRNG is a deterministic algorithm that produces numbers that appear to be random. It requires a seed value to generate a seemingly random number, and due to its deterministic nature, the same seed will produce the same value every time. PRNGs experience periodicity, for after exhausting all possible internal variations, it will repeat cycles that will reiterate on the sequences of produced numbers. Good PRNG algorithms, however, manage to display good statistical behaviors, with some having periods in order of magnitudes so large they become negligible. Due to their algorithmic nature, PRNGs are quick and scalable; their deterministic nature is also desirable in experiments where replicability is key. However, they are extremely unsafe key producers for cryptographic measures, for a backtrack of the algorithm or the knowledge of the seed reveal the output in its entirety (Mechalas, 2018). True Random Number Generators (TRNG), on the other hand, aim to produce true random values. To achieve this, TRNG relies on entropy sources, which extract true randomness by extracting information from physical phenomena. The two main types of entropy sources are dynamic entropy sources, which extract true random values from indeterminate physical processes like thermal noise or atmospheric noise, and static entropic sources, which extract randomness from randomly occurring properties in the hardware components of the computer as a result of the semiconductor manufacturing process, which become stable once the device is finished. These properties can be found in chip and are used for things like authentications (Cao et al., 2022). What is essential is that an entropy source extracts its randomness from the physical world, which means that no "randomness" generated by a procedural method can be considered an entropy source. TRNGs are desirable where safety is essential, and show constant distributions that guarantee unpredictability, but are usually slow to output a number due to their need to measure physical phenomena; this takes time and is computationally costly, which makes them not very scalable. Some methods can be classified into particular categories of RNG. Cascade Construction Random Number Generator (CCRNG), for example, relies on an entropy source to supply an "entropy buffer", which is then used to provide cryptographically secure PRNG. Digital Random Number Generators (DRNGs) is an approach that builds the RNG on the processor's hardware directly, and uses a combination of CCRNG and dynamic entropy sources to create random streams. Even so, this categories are usually some sort of combination of PRNG and TRNG to different degrees, and combining both methods is becoming more common in the industry to tackle the strengths and weaknesses of both methods.

III. Studied Methods

IV. Deep Learning Methods

Deep learning is a subset of machine learning that uses artificial neural networks with many layers to automatically learn complex patterns from large amounts of data.

V. techniques

VI. CTR_DRBG Counter mode Deterministic Random Bit Generator

CTR_DRBG (Counter mode Deterministic Random Bit Generator) is a standardized method for constructing a deterministic random bit generator (PRNG) using a block cipher operating in counter (CTR) mode. This technique is defined in NIST Special Publication 800-90A, titled *"Recommendation for Random Number Generation Using Deterministic Random Bit Generators"*. Essentially, CTR_DRBG transforms a secure symmetric cipher—such as AES—into a cryptographically strong source of pseudorandom bits. The counter mode ensures that each generated block is unique by systematically incrementing a counter value for each new data request, thus preventing the repetition of output sequences under the same key and seed. This approach is widely used in cryptographic applications requiring high security and reliability in random data generation, such as key generation, initialization vectors, and session tokens.

VII. Results

VIII. Conclusions

References