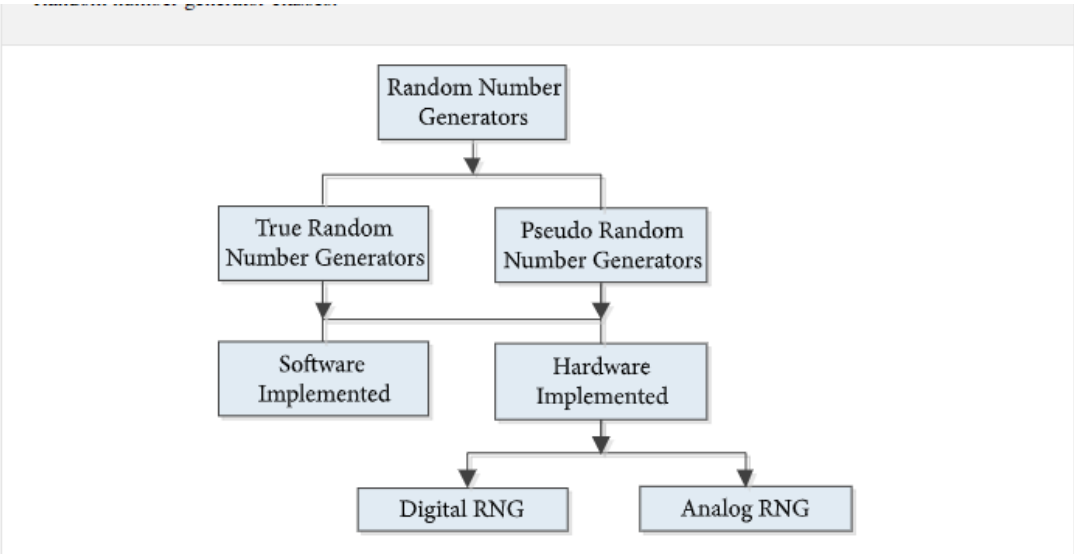
RANDOM NUMBER GENERATOR

- Random numbers are widely used in areas such as cryptography and data transmission, luck games, secure communication, simulation, and game programming, where key generation is important.
- **Random number generators can be divided into two classes:**
- TRNG (True Random Number Generator)
- PRNG (Pseudo Random Number Generator).
- Random numbers can be generated as hardware and software.
- The random numbers generated by the software can be defined by a specific mathematical model. On the other hand, it is possible to generate numbers by hardware with the help of noise source whose behavior cannot be predicted.



Applications of Random Number Generators Random Number Generators are used in a wide array of applications:

- Gaming
- Statistical Analysis
- Simulation
- Weather prediction
- Medicine
- Radio communications
- **Cryptography**

• For encryption key use, random number generators are used to create seed values (or starting values) from which the encryption algorithms will work.

Two Types of Random Number Generators:

- Pseudo-Random Number Generators (PRNG)
- True Random Number Generators (TRNG)

Pseudo-Random Number Generators (PRNG):

- **Pseudo Random Number Generator(PRNG)** refers to an algorithm that uses mathematical formulas to produce sequences of random numbers. PRNGs generate a sequence of numbers approximating the properties of random numbers.
- A PRNG starts from an arbitrary starting state using a **seed state**.
  Many numbers are generated in a short time and can also be reproduced later, if the starting point in the sequence is known. Hence, the numbers are **deterministic and efficient**.

NOTE:
Linear Congruential Generator is most common and oldest algorithm for generating pseudo-randomized numbers.

Characteristics of PRNG

- **Efficient:** PRNG can produce many numbers in a short time and is advantageous for applications that need many numbers
- **Deterministic:** A given sequence of numbers can be reproduced at a later date if the starting point in the sequence is known. Determinism is handy if you need to replay the same sequence of numbers again at a later stage.
- **Periodic:** PRNGs are periodic, which means that the sequence will eventually repeat itself. While periodicity is hardly ever a desirable characteristic, modern PRNGs have a period that is so long that it can be ignored for most practical purposes

Applications of PRNG

PRNGs are suitable for applications where many random numbers are required and where it is useful that the same sequence can be replayed easily. Popular examples of such applications are **simulation and modeling applications**. PRNGs are not suitable for applications where it is important that the numbers are really unpredictable, such as **data encryption and gambling.**

Pseudo Random Number Generator using srand()

True Random Number Generators (TRNG) :

True Random Number Generators TRNG's draw from the randomness of some accessible physical phenomena for generation of random numbers.

- **Examples** include radioactive decay, atmospheric noise, background (white) noise, and electrical or quantum phenomena.
- There is no discernable pattern in true random number generation and they can be counted on to produce something that is truly random.
- They are nondeterministic
- One would not be able to reproduce a given sequence of numbers.
- **TRNG's are better suited for encryption key generation**.

Note:

True random numbers are used for applications such as gaming, gambling, and in cryptography, where randomness is critically important. For example, many cryptographic algorithms and security protocols depend on keys and their strength is defined by the number of key bits that an attacker needs to determine before breaking a system. If keys are compromised, the security strength of the whole system is compromised.

True random numbers are required in a variety of security scenarios:

- Key generation for various algorithms (symmetric, asymmetric, MACs) and protocols (SSL/TLH, SSH, WiFi, LTE, IPsec, etc.)
- Chip manufacturing (seeding device unique and platform keys)
- Initial values (for encryption and MAC algorithms, TCP packet values, etc.)
- Nonce generation and initial counter values for various cryptographic functions
- Challenges used for protocol authentication exchanges
- Randomization input for side channel countermeasure solutions for protecting against physical attacks

TASK:

- **What is a TRNG and Why is It Important?**
- **comparison of PRNG and TRNG number generators**