# Criptografia e gerador de números pseudo-aleatórios

**Mirela Mei Costa**
mirelameic@usp.br

## Gerador de números pseudo-aleatórios

## Números aleatórios

Ao discutir números únicos, um número aleatório é aquele que é retirado de um conjunto de valores possíveis onde cada um é igualmente provável de ocorrer, ou seja, uma distribuição uniforme[1]. Em uma sequência de números aleatórios, cada número sorteado deve ser estatisticamente independente[2] dos outros.

Com o advento dos computadores, os programadores reconheceram a necessidade de um meio de introduzir aleatoriedade em um programa de computador. No entanto, por mais surpreendente que possa parecer, é difícil fazer com que um computador faça algo por acaso. Um computador segue suas instruções cegamente e, portanto, é totalmente previsível, um computador que não segue as instruções dessa maneira está danificado.

Existem duas abordagens principais para gerar números aleatórios usando um computador: Geradores de Números Pseudo-Aleatórios (PRNGs) e Geradores de Números Aleatórios Verdadeiros (TRNGs). As abordagens têm características bastante diferentes e cada uma tem seus prós e contras. Os números aleatórios e pseudo aleatórios gerados para aplicativos criptográficos devem ser imprevisíveis. No caso de PRNGs, se a semente (valor inicial) for desconhecida, o próximo número de saída na sequência deve ser imprevisível, apesar de qualquer conhecimento de números aleatórios anteriores na sequência. Esta propriedade é conhecida como imprevisibilidade direta. Também não deve ser viável determinar a semente a partir do conhecimento de quaisquer valores gerados (ou seja, imprevisibilidade para trás também é necessária).

[1] A distribuição uniforme discreta é uma distribuição de probabilidade simétrica em que um número finito de valores é igualmente provável de ser observado; cada um dos n valores tem probabilidade igual a 1/n.

[2] Dois eventos são estatisticamente independentes se a ocorrência de um não afeta a probabilidade de ocorrência do outro (equivalentemente, não afeta as probabilidades). Da mesma forma, duas variáveis aleatórias são independentes se a realização de uma não afetar a distribuição de probabilidade da outra.

## Geradores de Números Aleatórios Verdadeiros (TRNGs)

Os TRNGs podem extrair aleatoriedade de fenômenos físicos e introduzi-los em um computador, por meio de um hardware. Um fenômeno físico eficiente para usar provém das fontes radioativas; os momentos no tempo em que uma fonte radioativa decai são completamente imprevisíveis e podem ser facilmente detectados e alimentados em um computador, evitando quaisquer mecanismos de buffer do sistema operacional. Outro fenômeno físico adequado é o ruído atmosférico, que é bastante fácil de captar com um rádio normal. Independentemente de qual fenômeno físico é usado, o processo de geração de verdadeiros números aleatórios envolve a identificação de pequenas mudanças imprevisíveis nos dados.

Os TRNGs geralmente produzem apenas um número limitado de bits aleatórios por segundo, por isso são bastante ineficientes em comparação com os PRNGs, levando muito mais tempo para produzir a mesma quantidade de números. Eles também são não determinísticos, o que significa que uma dada sequência de números não pode ser reproduzida intencionalmente, embora a mesma sequência possa, é claro, ocorrer várias vezes por acaso. TRNGs não têm período, portanto a sequência não irá se repetir após uma quantidade n de números.

## Geradores de Números Pseudo-Aleatórios (PRNGs)

Como a palavra 'pseudo' sugere, os números pseudo-aleatórios não são verdadeiramente aleatórios. Essencialmente, os PRNGs são algoritmos que usam fórmulas matemáticas ou

simplesmente tabelas pré-calculadas para produzir sequências de números que parecem aleatórias. Um PRNG usa uma ou mais sementes e gera vários números "pseudo-aleatórios". Em contextos em que a imprevisibilidade é necessária, a própria semente deve ser aleatória e imprevisível. Portanto, por padrão, um PRNG pode obter suas sementes dos resultados de um TRNG.

A diferença básica entre PRNGs e TRNGs é como comparar números aleatórios gerados por computador e os lançamentos de um dado. Como os PRNGs geram números aleatórios usando fórmulas matemáticas ou listas pré-calculadas, usar um gerador desses corresponde a alguém jogar um dado muitas vezes e anotar os resultados. Sempre que você pede uma jogada de dados, você obtém o próximo na lista. Efetivamente, os números parecem aleatórios, mas são predeterminados. Os TRNGs funcionam fazendo com que um computador realmente role o dado - ou, mais comumente, use algum outro fenômeno físico que seja mais fácil de conectar a um computador do que um dado, como valores de ruído atmosférico ou algum outro fenômeno natural.

Os PRNGs são eficientes, o que significa que podem produzir muitos números em um curto espaço de tempo, e determinísticos, o que significa que uma sequência de números pode ser reproduzida em uma data posterior se o ponto de partida da sequência for conhecido. A eficiência é uma boa característica se o programa precisa de muitos números, e o determinismo é útil se precisar reproduzir a mesma sequência de números novamente em um estágio posterior. Além disso, os PRNGs normalmente são periódicos, o que significa que a sequência acabará se repetindo. Embora a periodicidade dificilmente seja uma característica desejável, os PRNGs modernos têm um período tão longo que pode ser ignorado para a maioria dos propósitos práticos - exemplos populares são os aplicativos de simulação e modelagem. Os PRNGs não são adequados para aplicativos em que é importante que os números sejam realmente imprevisíveis, como criptografia de dados e jogos de azar. A conclusão final é que, mesmo que um PRNG atenda às necessidades do programa em si, ainda é necessário ter cuidado onde usá-lo, pois podem ocasionar em erros e bugs indesejáveis.

**Definição matemática**
Dado:

$P$ – uma distribuição de probabilidades em $\mathbb{R}$ e $\mathcal{B}$ (onde $\mathcal{B}$ é o conjunto Borel padrão na linha real)

$\mathfrak{I}$ – uma coleção não vazia de conjuntos de Borel $\mathfrak{I} \subseteq \mathcal{B}$, por exemplo: $\mathfrak{I} = \{(-\infty, \mathrm{t}] : \mathrm{t} \in \mathbb{R}\}$. Se $\mathfrak{I}$ não é especificado, pode ser qualquer $\mathcal{B}$ ou $= \{(-\infty, \mathrm{t}] : \mathrm{t} \in \mathbb{R}\}$, dependendo do contexto.

$\mathrm{A} \subseteq \mathbb{R}$ – um conjunto não vazio (não necessariamente um conjunto de Borel). amiúde A é um conjunto entre $P$ suporte e seu interior; por exemplo, se $P$ é a distribuição uniforme no intervalo $(0,1]$, A pode ser $(0,1]$. Se A não é especificado, presume-se que seja algum conjunto contido no suporte de $P$ e contendo seu interior, dependendo do contexto.

Chamamos de função $\mathrm{f} : \mathbb{N}_1 \to \mathbb{R}$ (onde $\mathbb{N}_1 = \{1,2,3,\dots\}$ é o conjunto de inteiros positivos) um gerador de números pseudoaleatórios para $P$ dado $\mathfrak{I}$ tendo valores em A se e somente se $\mathrm{f}(\mathbb{N}_1) \subseteq \mathrm{A}$

$$\forall \mathrm{E} \in \mathfrak{I} \quad \forall 0 < \epsilon \in \mathbb{R} \quad \exists N \in \mathbb{N}_1 \quad \forall N \leq n \in \mathbb{N}_1, \left| \frac{\#\{i \in \{1,2,\dots,n\} : \mathrm{f}(i) \in \mathrm{E}\}}{n} - P(\mathrm{E}) \right| < \varepsilon$$

(#$S$ denota o número de elementos no conjunto finito $S$)

Pode-se mostrar que se f é um gerador de números pseudoaleatórios para a distribuição uniforme em $(0,1)$ e se $F$ é a CDF (função de distribuição cumulativa) de alguma distribuição de probabilidade dada $P$ então $F^* \circ \mathrm{f}$ é um gerador de números pseudoaleatórios para $P$ onde $F^* : (0,1) \to \mathbb{R}$ é o percentil de $P$, ou seja, $F^*(x) := \inf\{\mathrm{t} \in \mathbb{R} : x \leq F(\mathrm{t})\}$. Intuitivamente, uma distribuição arbitrária pode ser simulada a partir de uma simulação da distribuição uniforme padrão.

## Geradores de números pseudo-aleatórios criptograficamente seguros (CSPRNG)

Os CSPRNG são geradores de números pseudo-aleatórios (PRNG) com propriedades que os tornam adequados para uso em criptografia. Satisfazem o teste do próximo bit e resistem às extensões de comprometimento de estado e normalmente fazem parte do sistema operacional ou vêm de uma outra fonte externa e segura. Dependendo do nível de segurança necessário, o

CSPRNG pode ser implementado como componente de software, dispositivo de hardware ou como uma combinação de ambos. Os sistemas operacionais coletam entropia (semente inicial) do ruído ambiental: cliques do teclado, movimentos do mouse, atividade de rede, interrupções de E/S do sistema, atividade do disco rígido etc. Fontes de aleatoriedade do ambiente no Linux, por exemplo, incluem temporizações entre teclados, temporizações entre interrupções de algumas interrupções e outros eventos que são não determinísticos e difíceis de medir para um observador externo.

Os requisitos de um PRNG padrão também são satisfeitos por um PRNG criptograficamente seguro, porém o inverso não ocorre. Os requisitos do CSPRNG podem ser divididos em dois grupos: primeiro, eles devem passar por testes de aleatoriedade estatísticos; e segundo, eles devem ser resistentes a ataques, mesmo quando parte de seu estado inicial ou em execução chegue ao conhecimento do atacante. Todo CSPRNG deve satisfazer o teste do próximo bit. Isto é, dados os primeiros k bits de uma sequência aleatória, não há algoritmo de tempo polinomial capaz de predizer o (k+1)-ésimo bit com probabilidade de sucesso maior que 50%. Andrew Yao provou em 1982 que um gerador que passe nesse teste, passará em qualquer outro teste estatístico de tempo polinomial para aleatoriedade. Além disso, todo CSPRNG deve resistir a "extensões de compromisso de estado", ou seja, caso parte ou o todo de seus estados seja revelado (ou predito corretamente), deve ser impossível de reconstruir o fluxo de números aleatórios antes dessa revelação. Caso haja uma entrada de entropia em execução, deve ser inviável utilizar-se de qualquer conhecimento sobre o estado da entrada para predizer condições futuras sobre o estado do CSPRNG.

A maioria dos PRNGs não irão satisfazer ambos requisitos. Enquanto muitos PRNGs dão saídas aparentemente aleatórias para alguns testes estatísticos, eles não são capazes de resistir a determinados métodos de engenharia reversa. Testes estatísticos especializados podem ser especialmente melhorados para mostrar que esses números aleatórios não são verdadeiramente aleatórios. Além disso, para a maioria dos PRNGs, quando o seu estado atual é revelado, todos os números aleatórios passados podem ser calculados, permitindo a um atacante ler todas as mensagens passadas, bem como as futuras.

Na prática, há diversas formas de criá-los. Uma delas baseia-se em reunir uma sequência de *n* bits verdadeiramente aleatórios, onde *n* é um número grande o suficiente para impedir um ataque por força bruta, ou seja, um valor que seja inviável para tentar todas as *2^n* combinações de *n* bits. Com a tecnologia atual, um valor adequado seria n=128. Após isso, o sistema codifica esses eventos e os "comprime" aplicando uma função de hash criptograficamente segura, como SHA-256. A função hash, por sua definição, deve fazer um bom trabalho ao concentrar essa entropia em uma cadeia de bits *n* - bit.

## Eventos quânticos ou sistemas caóticos?

Uma característica que os construtores de TRNGs às vezes discutem é se o fenômeno físico usado é um fenômeno quântico ou um fenômeno com comportamento caótico. Há discordâncias sobre se os fenômenos quânticos são melhores ou não, e, ao final, tudo se resume às crenças sobre como o universo funciona. A questão chave é se o universo é determinístico ou não, ou seja, se tudo o que acontece é essencialmente predeterminado desde o Big Bang. O determinismo é um assunto difícil que tem sido objeto de muitas pesquisas filosóficas, e o problema está longe de ser solucionado.

A mecânica quântica é um ramo da física teórica que descreve matematicamente o universo nos níveis atômico e subatômico. Os geradores de números aleatórios baseados na física quântica usam o fato de que as partículas subatômicas parecem se comportar aleatoriamente em certas circunstâncias. Parece não haver nada que cause esses eventos e, portanto, muitos acreditam que eles não sejam determinísticos.

Em comparação, os sistemas caóticos são aqueles nos quais pequenas mudanças nas condições iniciais podem resultar em mudanças dramáticas no comportamento geral do sistema. Os sistemas climáticos são um bom exemplo disso, há também o conhecido efeito borboleta, um experimento mental no qual uma borboleta batendo as asas no Japão é capaz de afetar os ventos de maneira sutil, mas

critica o suficiente para causar um tornado no outro lado do globo.

Os defensores dos geradores de números aleatórios da variedade quântica argumentam que a física quântica é inerentemente não determinística, enquanto os sistemas governados pela física são essencialmente determinísticos. Há quem argumente que o ruído atmosférico usado como fonte para gerar números pode ser visto como um sistema caótico, mas determinístico. Portanto, se existir alguém que conheça o suficiente sobre os processos que causam ruído atmosférico (por exemplo, tempestades), essa pessoa poderia potencialmente prever os números gerados. No entanto, para fazer isso, provavelmente precisaria do conhecimento da posição e velocidade de cada molécula nos sistemas climáticos do planeta. Atualmente, isso é praticamente inviável, e a imprecisão das previsões do tempo é um bom exemplo de como é difícil dar até mesmo uma estimativa aproximada do comportamento dos sistemas meteorológicos.

Além disso, os geradores quânticos também não estão protegidos de críticas. Deterministas rígidos dizem que o comportamento das partículas subatômicas não é realmente aleatório, mas exatamente tão predeterminado como tudo o mais no universo é.A razão pela qual acredita-se que essas partículas específicas se comportam aleatoriamente é o simples fato de que nenhuma medição humana foi capaz de explicar seu comportamento. Nessa visão, os eventos subatômicos de fato têm uma causa anterior, mas nós simplesmente não a entendemos (ainda) e, portanto, os eventos parecem aleatórios para nós. Para um determinista rígido, a física quântica é exatamente tão adequada para a geração de números aleatórios quanto o ruído atmosférico.

Este é apenas um argumento possível dentre muitos outros. Portanto, até o momento, a definição mais significativa de aleatoriedade é aquela que não pode ser prevista por humanos. Se a aleatoriedade se origina de sistemas climáticos imprevisíveis, lâmpadas de lava ou eventos de partículas subatômicas, isso é amplamente acadêmico.

## Referências


Rukhin, Andrew et al. (2010)A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. National Institute of Standards and Technology Special Publication 800-22 revision 1a Natl. Inst. Stand. Technol. Spec. Publ. 800-22rev1a, 131 pages

Haahr, Mads. (1998) Introduction to Randomness and Random Numbers. Rambom.org

Sinharay, S. (2010). Discrete Probability Distributions. International Encyclopedia of Education, 132–134. Doi:10.1016/b978-0-08-044894-7.01721-

Tavares, Pedro. (2017) Pseudorandom Number Generator (PRNGS). Revista PROGRAMAR 57ª edição.

Wikipedia. (2012). CSPRNG. Disponível em: https://pt.wikipedia.org/wiki/CSPRNG

Baldanzi, L., Crocetti, L., Falaschi, F., Bertolucci, M., Belli, J., Fanucci, L., & Saponara, S. (2020). Cryptographically Secure Pseudo-Random Number Generator IP-Core Based on SHA2 Algorithm. Sensors, 20(7) 1869. Doi:10.3390/s20071869

Nakov, Svetlin. (2019). Secure Random Generators (CSPRNG). Practical Cryptography for Developers. Disponível em: https://cryptobook.nakov.com/secure-random-generators/secure-random-generators-csprng