



Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

SECURITY

Zufallszahlen und -generatoren

May 31, 2024

Marc Stöttinger



Random numbers should not be generated with a method chosen at random.

Donald Knuth

ZUFALLSZAHLENGENERIERUNG

- Kryptographische Verfahren benötigen sichere Zufallszahlen für
 - Wahl symmetrischer Schlüssel
 - Initialisierungsvektoren (IVs) für Betriebsmodi von Blockchiffren
 - Primzahl- und Parametergenerierung bei RSA, DSA, DH, ...
- Aber was sind „sichere“ Zufallszahlen?
 - Welche Eigenschaften müssen „sichere“ Zufallszahlen haben?
 - Wie sieht ein „sicherer“ Zufallszahlengenerator aus?

DISKUSSION IN KLEINEN GRUPPEN

Welche der folgenden Zufallszahlenfolgen sind „sicher“?

- 42 42 42 42 42 42 42 42 42 42
- 1 6 11 16 21 26 31 36 41 46 51
- 3141 5926 5358 9793 2384

Eigenschaften von kryptographischen Zufallszahlen

Welche Eigenschaften machen „sichere“ Zufallszahlen aus?

BEISPIELE FÜR UNSICHERE ZUFALLSZAHLENGENERATOREN

Welche der folgenden Zufallszahlenfolgen sind „sicher“?

- 42 42 42 42 42 42 42 42 42 42 → Kein Zufall
- 1 6 11 16 21 26 31 36 41 46 51 → Nicht gleichverteilt
- 3141 5926 5358 9793 2384 → Vorhersagbar! Stellen von Pi sind bekannt

Eigenschaften von kryptographischen Zufallszahlen

Welche Eigenschaften machen „sichere“ Zufallszahlen aus?
Gleichverteilung, Nichtvorhersagbarkeit

WAS FÜR EIGENSCHAFTEN HABEN ZUFALLSZAHLEN?

Ideale Zufallszahlen sind **nichtvorhersehbar**, **unabhängig** und **gleichverteilt** (ideale Zufälligkeit).

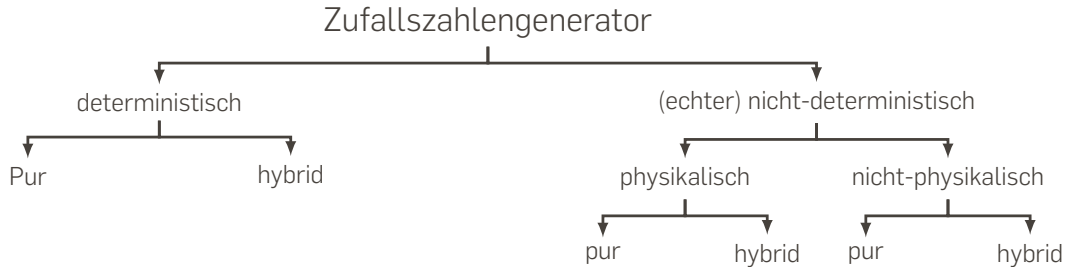
- Nichtvorhersehbarkeit: Die Zufallszahl ist vor der Generierung mit einem gewissen Grad der Ungewissheit nicht vorhersehbar. Der Grad der Ungewissheit kann durch die Entropie quantifiziert werden.
- Unabhängigkeit: Die Erzeugung früherer Zufallszahlen darf keinen nachvollziehbaren Einfluss auf die aktuelle Erzeugung einer Zufallszahl haben.
- Gleichverteilung: Jeder zulässige Wert einer Zufallszahl hat die gleiche Chance, einzutreten.

Entropie

Eine gute Entropiequelle ist essentiell für einen Zufallszahlengenerator.

KRYPTOGRAPHISCHER ZUFALLSZAHLENGENERATOREN

- Ein RNG besteht aus einem **nicht-deterministischen** Teil (Entropiequelle) und einem **deterministischen Teil**, der aus diesen Daten die Ausgangssequenz des RNG (Zufallszahlen) erzeugt.
- Der nicht-deterministische Teil des RNG nutzt eine **physikalische Entropiequelle** oder **nicht-physikalische Entropiequelle** zur Erzeugung einer Zufallszahlenfolge.



ECHE ZUFALLSZAHLENGENERATOREN (PTRNG)

- Der Kern eines jeden **Physical True Random Number Generator** (PTRNG) ist die **Entropiequelle** → **raw** Zufallszahlen.
 - Durch Ausnutzung eines analogen Signals erzeugt ein **Digitalisierungsmechanismus** eine Folge von **digitalen "rohen" Daten**
 - Ein **Nachbearbeitungsalgorithmus** wandelt die Rohdaten in **interne Zufallszahlen** um.
- **Zeitdiskrete** physikalische Entropiequelle ist z.B.,
 - Radioaktiver atomarer Zerfall
- **Analog** Physikalische Entropiequellen sind z.B.,
 - Thermische Widerstandsentropie: Die Spannung zwischen Widerständen schwankt zufällig aufgrund der Vibration von Atomen.
 - Diodendurchbruch-Entropie: Der Sperrstrom durch Dioden variiert zufällig aufgrund des Tunnelns von Elektronen.

BEISPIEL FÜR ECHTE ZUFALLSZAHLENGENERATOREN (TRNG)

- Basieren auf **nichtvorhersehbaren, physikalischen** Prozessen
 - von bewegten Klumpen in Lavalampen
 - Sperrstrom durch Dioden schwankt zufällig aufgrund von Tunneln von Elektronen
- Statistische Verteilung des Rauschens ist **häufig suboptimal**
 - Lösung: TRNG aggregiert Messungen, um gute Verteilung zu erhalten
 - Ausgaberate des TRNGs entsprechend niedrig (\sim KB/sec)



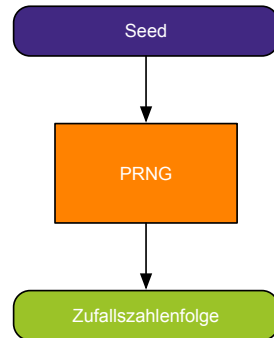
Quelle: <https://www.cloudflare.com/de-de/learning/ssl/lava-lamp-encryption/>

ECHTE ZUFALLSZAHLENGENERATOREN (NPTRNG)

- Ein **nichtphysikalischer echter Zufallszahlengenerator** (NPTRNG) verwendet **externe Signale** als Entropiequelle.
- Das Konzept der Zufälligkeit ist der **Mangel an Information** über Prozesse und ihre Ergebnisse.
- Externe Entropiequellen sind z.B.,
 - **Prozesse** wie Platten-I/O-Operationen und Interrupts (z.B. Linux RNG **/dev/random**).
 - **Systemdaten** als Tickzähler seit Systemstart, Prozess- und Thread-IDs, und aktuelle Ortszeit (z.B. in MS Windows CE Enhanced Cryptographic Provider).
 - **Menschliche Interaktion** als Mausbewegung und Tastenanschläge (z.B. PGP-Schlüsselerzeugung).
- Eine große Menge von Daten aus verschiedenen Quellen und Nachbearbeitung (z.B. durch eine Hash-Funktion) sind erforderlich.

PSEUDO ZUFALLSZAHLENGENERATOREN (PRNG)

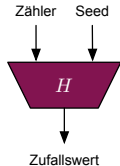
- Algorithmus, der aus einem Startwert (Seed) eine Zufallszahlenfolge erzeugt
 - Deterministisch: Gleicher Seed → Gleiche Zufallszahlenfolge
- Sicherheit eines PRNG hängt ab von
 - Zufälligkeit und Nichtvorhersagbarkeit des Seeds
 - Verwendetem Algorithmus
- Brute-Force auf Seed ist möglich, da PRNG deterministisch
 - Mindestens 2^{128} Möglichkeiten für Seed (für Rechnerische Sicherheit)
- Empfohlen Standard für Anforderungen an den Seed [SP800-90ARev1] und [SP800-90C3pd]



EIGENSCHAFTEN EINER PRNG FUNKTION

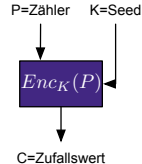
→ Sicherheitseigenschaften einer PRNG

1. Vom Zufallswert darf nicht auf den Seed geschlossen werden
→ Ansonsten: Berechnung des Seeds und Bruch der Nichtvorhersagbarkeit
2. Der Zufallswert muss gut verteilt sein
→ Ansonsten: Ausnutzung einer schlechten Verteilung



Sicherheitseigenschaften:

1. Einwegfunktion
2. Starke Kollisionsresistenz



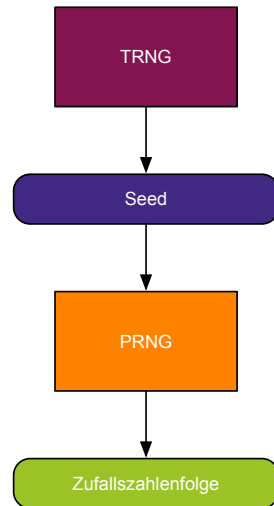
Sicherheitseigenschaften:

1. Seed nicht errechenbar aus C und P
2. eindeutige Zuordnung durch $C = Enc_K(P)$

→ Empfohlener Standard für Konstruktion von PRNGs [SP800-90ARev1]

HYPRIDE ZUFALLSZAHLENGENERATOREN

- Kurzer Seed wird aus TRNG generiert
 - Umgeht Problem der langsamen TRNG Geschwindigkeit
- Seed dient als Eingabe zum PRNG und wird dort beliebig erweitert
- Kombiniert die Vorteile beider Verfahren
 - Sicherheit basiert auf echtem Zufall
 - Effizienz von PRNG wird genutzt
- Empfohlener Standard für Konstruktion [AIS].



PROBLEME BEI ZUFALLSZAHLENGENERATOREN

- **Grundproblem**: Nichtvorhersagbarkeit und Gleichverteilung nur schwer prüfbar
- Häufige Sicherheitsprobleme bei Zufallszahlengeneratoren
 - **Konstanter Wert** oder **Uhrzeit** als Seed für PRNG (z.B.: MiFare Chips [MIF])
 - **TRNG** hat **schlechte Verteilung** (z.B.: gemeinsame Primzahl in RSA)
 - **PRNG Eigenkonstruktionen** mit schlechtem Design (z.B.: Windows PRNG [DGP07])
 - **Implementierungsfehler** in PRNG (z.B.: Android Java PRNG [MMS13])
- Möglichkeiten zur Überprüfung
 - Statistische Tests für Gleichverteilung (DieHarder Testsuite, TestU01, ...)
 - BSI Standardisierungen für TRNGs/PRNGs (AIS 20 und 31 Standards [AIS])

ZUSAMMENFASSUNG

- Eigenschaften kryptographischer Zufallszahlengeneratoren
- Eigenschaften und Unterschiede von TRNGs, PRNGs und hybriden RNGs
- Beurteilen ob gewählte Funktionen die Eigenschaften von kryptographischer Zufallszahlengeneratoren erfüllt