

# Zufallsgenerator = Achillesferse

```
110000011110111011111111111110000011  
0110010101100100111111111010110  
011010100011001000001110010000  
101101000011110111101011100000  
010111100010101100111000101111  
111100000110100010100111011110  
100111101010110010000000001000  
110101010001001110000110011010  
011001001101011
```

Bootvorgang, ID

Mail, Message-ID

Web-Browser

Spiel

**danach:** Kryptoschlüssel

# Zufall

Unvorhersagbarkeit

Gegenteil von berechenbar

*computationally infeasible to predict*

# Zufallsgeneratoren prüfen

statistical test suite (description)

[csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf](https://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf)

Hypothesis  $H_0$ : the sequence being tested is random

falscher Alarm (Typ-I-Error) mit Wahrscheinlichkeit  $< 1 \%$

# Zufallsgeneratoren prüfen

Frequency (Monobit) Test

Frequency Test within a Block

Runs Test

Test for the Longest Run of Ones in a Block

Binary Matrix Rank Test

Discrete Fourier Transform (Spectral) Test

Non-overlapping Template Matching Test

Overlapping Template Matching Test

Maurer's „Universal Statistical“ Test

Linear Complexity Test

Serial Test

Approximate Entropy Test

Cumulative Sums (Cusum) Test

Random Excursions Test

Random Excursions Variant Test

# Hilfsmittel

- Gamma function

$$\Gamma(t) = \int_0^{\infty} x^{t-1} e^{-x} dx$$

$$\Gamma(n) = (n - 1)!$$

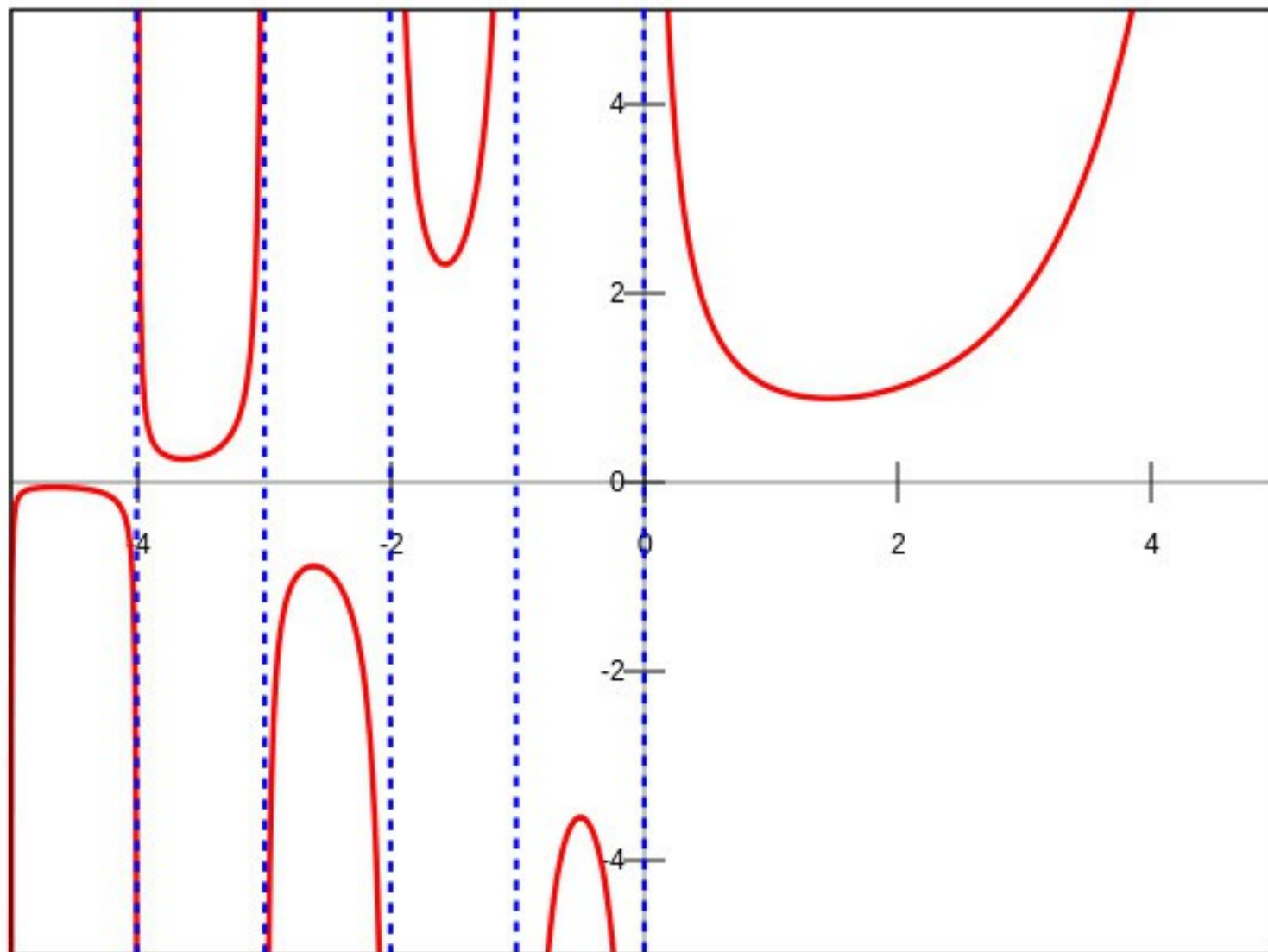
$$\Gamma(a, x) = \int_x^{\infty} t^{a-1} e^{-t} dt$$

GSL-Library (GNU scientific library)

```
double gsl_sf_gamma_inc_Q (double a, double x)
```



# Gamma function



# Hilfsmittel

- Error function  
*erfc(x)*

- in C

- in Python

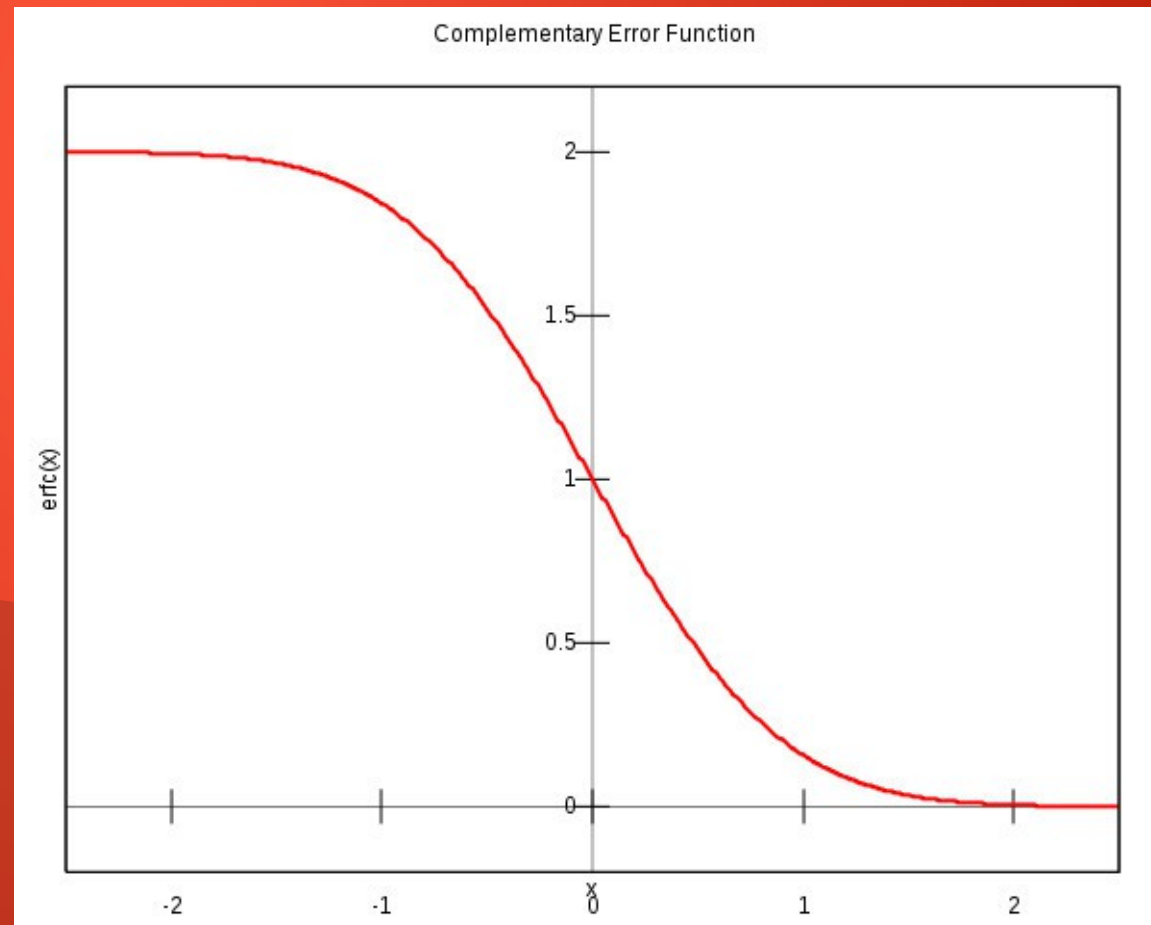
Normalverteilung:

Mittelwert,

Standardabweichung

prüft, ob Abweichung zu groß

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$$





# $\chi^2$ -Test

- Aufteilung in Kategorien
- Stelle fest, ob Auftreten der Fälle zufällig
- Freiheitsgrade  $k$  : wieviele Fälle frei wählbar

$$d_i = \frac{(c_i - e_i)^2}{e_i}$$

$$\chi^2 = \sum_i d_i$$

$$P = \Gamma(k/2, \chi^2/2)$$

# Monobit-Test

- $f(\text{bit})=2^{\text{bit}}-1$ ,  $n = \text{Anzahl der Bits}$
- Summe
- Mittelwert 0
- Abweichung von 0 mit Errorfunktion  $\text{erfc}()$

$$s = \frac{|S|}{\sqrt{n}}$$

$$P = \text{erfc} \left( \frac{s}{\sqrt{2}} \right)$$

# Frequency in Block

- N Blöcke der Größe M
- $N = n / M$
- $M > 0.01 * n > 20$ ,  $N < 100$
- Prozentsatz von 1en in i-tem Block

$$\chi^2 = 4M \sum_{i=1}^N \left( \pi_i - \frac{1}{2} \right)^2$$

$$P = \Gamma(N/2, \chi^2/2)$$

# Runs

- 01111.....10 ist ein 1-run
- wie oft aufeinanderfolgende Bits nicht identisch

$$V_n = \sum_{i=0}^{n-2} (\epsilon_i \neq \epsilon_{i+1})$$

$$P = \text{erfc} \left( \frac{|V_n - 2n\pi(1 - \pi)|}{2\sqrt{2n\pi(1 - \pi)}} \right)$$

Variante: blockweise, Blöcke der Länge 8, 128, 10000

# Matrix Rank

- 32 x 32 Bits als Matrix
- Rang der Matrix mit prozentualer Erwartung
  - Voller Rang in 28% aller Fälle
  - Rang 31 in 58% der Fälle
  - Rang <31 in 13% der Fälle
- Drei Klassen →  $\chi^2$ -Test

$$P = e^{-\chi^2/2}$$

# Fourier-Transformation

- $f(b)=2*b-1$

$$a_j = \sum_{k=0}^{n-1} f(\epsilon_k) \cdot \exp(2\pi i \cdot k \cdot j/n)$$

$$\exp(2\pi i \cdot k \cdot j/n) = \cos(2\pi k \cdot j/n) + i \sin(2\pi k \cdot j/n)$$

$$|a| = \sqrt{c^2 + d^2}.$$

# Template Matching

- feste Schablone von Bits, z.B. 10111
- über die Folge gleiten lassen, wie oft ein Treffer?
- Überlappend / nicht-überlappend

# Maurer Universal Entropy

- Wie gut lässt sich die Folge mit Komprimierungsverfahren (Ziv) behandeln?
- Wieviel Information enthält die Folge?



# Linear Complexity

- Finde LFSR-Rekursionsformel für Block der Länge  $M$
- Länge des LFSR, Erwartungswert

$$\mu = \frac{M}{2} + \frac{9 + (-1)^{M+1}}{36} - \frac{\frac{M}{3} + \frac{2}{9}}{2^M}$$

$$T_i = (-1)^M \cdot (L_i - \mu) + \frac{2}{9}$$

class $k$	condition	$\pi_k$
0	$T_i \leq -2.5$	0.010417
1	$-2.5 < T_i \leq -1.5$	0.031250
2	$-1.5 < T_i \leq -0.5$	0.125000
3	$-0.5 < T_i \leq +0.5$	0.500000
4	$+0.5 < T_i \leq +1.5$	0.250000
5	$+1.5 < T_i \leq +2.5$	0.062500
6	$+2.5 < T_i$	0.020833

# Random Excursions

- $f(b) = 2^b - 1$
- Addiere die  $f()$ -Werte, drei Tests
  - Wie groß ist der größte Abstand von 0?
  - Wie oft wird 0 durchquert?
  - Wie oft kommt Zustand  $s$  vor  $\{-9, -8, \dots, -1, 1, 2, \dots, 9\}$

# Beispiel: DUAL-EC DBRG

- bis 2006: keine Standards für Zufallsgeneratoren
- in 2006: Vorschlag mit 4 Algorithmen
- Shumow/Ferguson (Microsoft), CRYPTO 2007, mögliche NSA-Hintertür im DUAL-EC-Verfahren
- standardisierte Parameterwahl nicht erläutert



# DUAL-EC (vereinfacht)

- (1)  $a=2$ ,  $b=3$ ,  $p=13$  feste Werte
- (2) wähle einen Startwert  $x$
- (3) berechne  $y = (a \cdot x) \bmod p$
- (4) berechne  $z = (b \cdot y) \bmod p$
- (5) Zufallsbits: unterste 4 Bits von  $z$
- (6) setze  $x=y$  und gehe zu (3)



Zufallsfolge mit  $x=10$  :      0, 3, 6, 4, 3, 1, 5, 2, 7, ....

# DUAL-EC (vereinfacht)

(1)  $a=2$ ,  $b=3$ ,  $p=13$  feste Werte

(2) wähle einen Startwert  $x$

$$x=10$$

(3) berechne  $y = (a \cdot x) \bmod p$

$$2 \cdot 10 = 20 \bmod 13 = 7$$

(4) berechne  $z = (b \cdot y) \bmod p$

$$7 \cdot 3 = 21 \bmod 13 = 8$$

(5) gib die untersten 3 Bits von  $z$  aus

$$8 = (1000)_2$$

(6) setze  $x=y$  und gehe zu (3)

Beispiel: Startwert  $x=10$

Zufallszahlen:

0, 3, 6, 4, 3, 1, 5, 2, 7, ....

man muss nur raten ob hier eine 0 oder eine 1 steht

$$0 = (0000)_2$$

# DUAL-EC-Trick der NSA

1111101111011100 01110100000010110010010...1010011101111010011110101

65536 Möglichkeiten

16 geheime Bits  
im Zustand des  
Zufallszahlenerzeugers

240 ausgegebene  
Zufallsbits

(relativ) viele ausgegebenen Zufallsbits → Anfängerfehler

besser: viele Bits im geheimen Zustand → weniger Performance

# Eine 10 Millionen \$ Frage

## RSA BSAFE

CRYPTOGRAPHIC, CERTIFICATE, AND TRANSPORT  
LAYER SECURITY (TLS) SOLUTIONS

RSA

- **NSA HQ:** würden Sie DUAL-EC als Voreinstellung in BSAFE einbauen?
- **RSA Inc:** Nein!
- **NSA HQ:** und wenn wir Ihnen 10 Millionen \$ geben?
- **RSA Inc:** Okay.

[www.reuters.com](http://www.reuters.com) am 21.12.2013

**BACKDOORED 2004-2013**