

# ECHTER ZUFALL

*Seminar Zufall ist Absicht*

*Blömer & von zur Gathen & May & Müller & Nüsken*

KRISTIAN WILKENING

Sommersemester 2001

**Zusammenfassung.** Die Seminararbeit führt ein in die Gewinnung von echtem Zufall für kryptographische Anwendungen. Sie zeigt, dass viele physikalische Effekte gut zur Zufallsgewinnung geeignet sind, und behandelt eine Reihe von algorithmischen Techniken, die aus „rohem“ Zufall echten Zufall erzeugen können. Weiterhin werden Gefahren im Zusammenhang mit der Zufallserzeugung und -verarbeitung aufgezeigt sowie Möglichkeiten vorgestellt, diese Gefahren zu vermeiden. Verschiedene Beispiele illustrieren schließlich, wie Zufall erzeugende Systeme aussehen können.

## 1. Einleitung

Die heutigen rechnergestützten Sicherheitssysteme bauen auf immer stärkeren Kryptographiealgorithmen auf, die nicht mittels Tricks wie Musteranalyse zu knacken sind. Die Sicherheit dieser Systeme hängt jedoch von den geheimen Werten für Passwörter, Schlüssel und ähnliches ab. Das Herz aller kryptographischen Systeme ist die Erzeugung geheimer, nicht zu erratender (also zufälliger) Zahlen. Die Verwendung pseudozufälliger Prozesse zur Erzeugung geheimer Werte kann in Pseudosicherheit resultieren. So mag es für den Angreifer dieser Sicherheitssysteme einfacher sein, die Umgebung zu reproduzieren, die die geheimen Werte hervorgebracht hat, und die daraus resultierende kleine Menge von Möglichkeiten zu durchsuchen, als die geheimen Werte im gesamten Zahlenraum zu suchen.

## 2. Motivation

Pseudozufallszahlgeneratoren führen iterativ mathematische Operationen auf einem Wert aus; ein Teil dieses Wertes stellt die pseudozufällige Ausgabe der jeweiligen Iteration dar. Der initiale Wert, auf dem die erste Iteration ausgeführt wird, heißt Saat. Die Ausgabe guter Pseudozufallszahlgeneratoren ist nicht vorhersagbar - solange man ihre Saat nicht kennt. Dies ist der Vorteil

von echtem Zufall gegenüber dem Pseudozufall: Er ist absolut unvorhersagbar.

Die Gewinnung von echtem Zufall ist jedoch um ein Vielfaches zeitaufwändiger als die Erzeugung von Pseudozufallszahlen. Dadurch bleiben Sicherheitsanwendungen, die ständig eine große Menge Zufall benötigen, auf Pseudozufallszahlgeneratoren angewiesen, während Anwendungen mit niedrigerem Zufallsbedarf auch mit echtem Zufall bedient werden können. Sicherheitsanwendungen, die nur selten bzw. nur einmalig eine bestimmte Menge Zufall benötigen, sind zum Beispiel die Erzeugung von kryptographischen Schlüsseln oder die Initialisierung von Pseudozufallszahlgeneratoren mit einer (echt) zufälligen Saat.

### 3. Physikalischer Zufall

Echter Zufall lässt sich zuverlässig aus verschiedenen physikalischen Effekten gewinnen, die sich entweder erwiesenermaßen chaotisch verhalten oder deren zu Grunde liegendes Modell nicht auf einem Rechnersystem modellierbar ist. Beispiele solcher physikalischen Effekte sind der Zerfall radioaktiver Stoffe, Kriechströme in Halbleitern [Agnew (1987)], Periodenschwankungen bei Oszillatoren [Fairfield *et al.* (1984)] sowie Luftturbulenzen in Festplatten [Davis *et al.* (1994)]. Auf diese Effekte wird (mit Ausnahme des radioaktiven Zerfalls) später näher eingegangen.

### 4. Roher Zufall vs. Echter Zufall

**4.1. Gestalt physikalischen Zufalls.** Alle physikalischen Zufallsquellen haben gemeinsam, dass der aus ihnen zu gewinnende Zufall nicht gleichverteilt ist, also nach dem Übersetzen der Effekte in binäre Informationen die Wahrscheinlichkeit für eine „1“ immer größer (beziehungsweise kleiner) als die Wahrscheinlichkeit für eine „0“ ist. Zudem sind die sukzessive gewonnenen Bits oftmals nicht vollständig unabhängig voneinander und dadurch in gewissem Maße vorhersagbar [Fairfield *et al.* (1984)]. Man kann also bei physikalischem Zufall von „rohem Zufall“ sprechen, aus dem noch durch geeignete Methoden ein kryptographischen Anforderungen gerecht werdender (also gleichverteilter und unvorhersagbarer) echter Zufall gewonnen werden muss.

**4.2. Entzerren.** Die Ungleichverteilung des rohen Zufalls lässt sich durch verschiedene Techniken des Entzerrens (engl. de-skewing) aufheben. Dabei wird davon ausgegangen, dass in einem zufälligen Wert nur eine bestimmte Menge an tatsächlich zufälliger *Information* (im informationstheoretischen

Sinn) vorhanden ist, wie Shannon (1948) zeigt:

Gegeben sei ein Satz von von  $n$  möglichen Ereignissen, deren Wahrscheinlichkeiten, dass sie auftreten,  $p_1, p_2, \dots, p_n$  sind, also eine diskrete Verteilung. Die sogenannte Entropie  $H$  ist dann ein Maß für die Wahlfreiheit bei der Auswahl eines der  $n$  Ereignisse. Genauer gesagt bezeichnet  $H$  die Anzahl von Bits an Information, die die Verteilung enthält.

$$H = - \sum_{i=1}^n p_i \cdot \log_2 p_i$$

Eine physikalische Zufallsquelle, die beispielsweise die 4 möglichen Werte 1, 2, 3 und 4 mit den Wahrscheinlichkeiten  $p(„1“) = \frac{1}{8}$ ,  $p(„2“) = \frac{1}{8}$ ,  $p(„3“) = \frac{1}{4}$  und  $p(„4“) = \frac{1}{2}$  ausgeben kann, bringt also nicht etwa  $\log_2 4 = 2$  Bits an Information pro ausgegebenem Wert, sondern nur  $-(\frac{1}{8} \cdot (-3) + \frac{1}{8} \cdot (-3) + \frac{1}{4} \cdot (-2) + \frac{1}{2} \cdot (-1)) = \frac{7}{4}$  Bits. Das angestrebte Ideal ist natürlich die Gleichverteilung, also  $p_1 = p_2 = \dots = p_n = \frac{1}{n}$ . In dem Fall ist  $H = \log_2 n$ . Um die Gleichverteilung zu erreichen, muss man eine genügend große Menge rohen Zufall nehmen und daraus ein Bit gleichverteilten Zufall berechnen. Dazu verwendet man eine der im Folgenden beschriebenen Techniken.

**4.2.1. Entzerren durch Parität.** Die Paritätsfunktion bildet einen Bitstring der Länge  $n$  entweder auf „0“ oder „1“ ab, und zwar je nachdem, ob der Bitstring eine gerade oder ungerade Anzahl Einsen aufweist. Die Paritätsfunktion einer nicht gleichverteilten roh zufälligen Eingabe ist nicht exakt gleichverteilt, kann aber beliebig nah an die Gleichverteilung angenähert werden [Eastlake *et al.* (1994)]. Voraussetzung dafür ist natürlich, dass die Zufallsquelle sowohl Bitstrings mit einer geraden als auch solche mit einer ungeraden Anzahl Einsen hervorbringt. Wieviele Bits nötig sind, um mittels Parität aus einem roh zufälligen Bitstring ein Bit „beinahe echten“ Zufalls zu erlangen, hängt vom Maß der Ungleichverteilung im rohen Zufall sowie der gewählten Schranke für die maximale Abweichung der Ausgabe von der Gleichverteilung ab.

Die Wahrscheinlichkeit, dass die Parität von  $n$  aufeinanderfolgenden rohen Zufallsbits (also eines  $n$ -Bitstrings) eine „1“ ergibt, ist die Summe der „geraden“ beziehungsweise „ungeraden“ Fälle (je nachdem, ob  $n$  gerade oder ungerade ist) in der Binomialverteilung von  $(p+q)^n$ , wobei  $p = \frac{1}{2} + e$  die Wahrscheinlichkeit für eine „1“ und  $q = \frac{1}{2} - e$  entsprechend die Wahrscheinlichkeit für eine „0“ in der Eingabe ist;  $e$  ist hierbei ein Maß für die Abweichung von der Gleichverteilung. Die Summen  $s_g$  der „geraden“ und  $s_u$  der „ungeraden“ Fälle in der Binomialverteilung kann man wie folgt ausrechnen ( $k$  ist hierbei jeweils die Anzahl Einsen):

$$(p+q)^n = \sum_{k=0}^n \binom{n}{k} p^k q^{n-k} = \underbrace{\left( \sum_{\substack{k=0 \\ k \text{ gerade}}}^n \binom{n}{k} p^k q^{n-k} \right)}_{=:s_g} + \underbrace{\left( \sum_{\substack{k=0 \\ k \text{ ungerade}}}^n \binom{n}{k} p^k q^{n-k} \right)}_{=:s_u}$$

$$(p-q)^n = \sum_{k=0}^n \binom{n}{k} p^k (-q)^{n-k} = \begin{cases} s_g - s_u & \text{falls } n \text{ gerade} \\ -s_g + s_u & \text{falls } n \text{ ungerade} \end{cases}$$

Daraus ergibt sich für ein gerades  $n$

$$\begin{aligned} s_g &= \frac{1}{2} \cdot ((p+q)^n + (p-q)^n) \\ s_u &= \frac{1}{2} \cdot ((p+q)^n - (p-q)^n) \end{aligned}$$

bzw. für ein ungerades  $n$  genau umgekehrt

$$\begin{aligned} s_g &= \frac{1}{2} \cdot ((p+q)^n - (p-q)^n) \\ s_u &= \frac{1}{2} \cdot ((p+q)^n + (p-q)^n) \end{aligned}$$

Da  $p+q = 1$  und  $p-q = 2e$ , reduzieren sich diese Ausdrücke bei geradem  $n$  auf

$$\begin{aligned} s_g &= \frac{1}{2} \cdot (1 + (2e)^n) \\ s_u &= \frac{1}{2} \cdot (1 - (2e)^n) \end{aligned}$$

bzw. bei ungeradem  $n$  auf

$$\begin{aligned} s_g &= \frac{1}{2} \cdot (1 - (2e)^n) \\ s_u &= \frac{1}{2} \cdot (1 + (2e)^n) \end{aligned}$$

Damit diese Wahrscheinlichkeiten nicht mehr als  $d$  von der Gleichverteilung abweichen, muss gelten

$$\frac{1}{2} \cdot (2e)^n < d$$

Durch Auflösen nach  $n$  erhält man

$$n > \log_{2e} 2d$$

In der Praxis bedeutet das zum Beispiel, dass man bei einer Unleichverteilung des rohen Zufalls von  $e = 0,3$  (also einer 80:20 Verteilung) und einer gewählten Genauigkeit der Ausgabe von  $d = 0,001$  nur  $n = 13$  Bits benötigt, um daraus ein Bit beinahe echten Zufalls zu erzeugen. (Beinahe,

da der tatsächliche *Informations*gehalt dieses Bits etwas weniger als ein Bit beträgt.) Die Paritätsfunktion lässt sich hardwareseitig leicht mittels eines XOR-Gatters (Exklusiv-Oder-Verknüpfung) realisieren.

**4.2.2. Entzerren durch Übergangsabbildung.** Die Übergangsabbildung geht zurück auf von Neumann (1963) und unterteilt den Eingabe-Bitstring in nicht überlappende Bitpaare. Für jedes Bitpaar gibt es vier mögliche Werte mit, je nach Verteilung des Bitstrings, teilweise unterschiedlichen Wahrscheinlichkeiten (siehe Tafel 4.1);  $e$  ist auch hier wieder ein Maß für die Abweichung von der Gleichverteilung. Da die Wahrscheinlichkeiten von „01“ und „10“ exakt übereinstimmen, wird „01“ auf die „0“ abgebildet und „10“ auf die „1“; die Paare „00“ sowie „11“ werden einfach ignoriert, d.h. sie liefern überhaupt keine Ausgabe.

Bitpaar	Wahrscheinlichkeit
00	$(\frac{1}{2} - e)^2 = \frac{1}{4} - e + e^2$
01	$(\frac{1}{2} - e) \cdot (\frac{1}{2} + e) = \frac{1}{4} - e^2$
10	$(\frac{1}{2} + e) \cdot (\frac{1}{2} - e) = \frac{1}{4} - e^2$
11	$(\frac{1}{2} + e)^2 = \frac{1}{4} + e + e^2$

Tafel 4.1: Bitpaare bei der Übergangsabbildung

Wie man sieht, liefert diese Methode im Gegensatz zur Parität eine exakte Gleichverteilung. Ein Nachteil ist die zufällige Laufzeit dieses Verfahrens, denn man kann nicht berechnen sondern nur abschätzen, wieviele Bits eingelesen werden müssen, um ein Bit auszugeben. Diese Ungewissheit bewegt sich jedoch in einem vertretbaren Rahmen, da beispielsweise bei einer 80:20 Verteilung im Durchschnitt nur 6,25 Bits benötigt werden, um ein Bit echten Zufalls auszugeben; die Wahrscheinlichkeit, dass 100 Bits für ein ausgegebenes Bit notwendig werden, beträgt in diesem Beispiel weniger als  $2 \cdot 10^{-9}$ .

**4.2.3. Weitere Entzerrungstechniken.** Neben den beschriebenen lassen sich noch eine Reihe weiterer Verfahren zur Entzerrung nutzen. Es seien hier die diskrete Fourier-Transformation nach Brillinger (1981) und Datenkompressionsalgorithmen genannt. Die Datenkomprimierung eignet sich zur Entzerrung, da sie darauf ausgerichtet ist, bei gleichbleibender Informationsmenge die Datenmenge zu reduzieren, also die *Information pro Bit* (siehe Shannon 1948) zu erhöhen. Allen Entzerrungstechniken gemeinsam ist die unvermeidliche Verkleinerung der Datenmenge bei gleichzeitig verbesserter Wahrscheinlichkeitsverteilung.

## 5. Gefahren

Bei der Gewinnung von echtem Zufall lauern eine Reihe von Gefahren, die es durch einen klugen und robusten Systementwurf zu vermeiden gilt. Im Gegensatz zu den Pseudozufallszahlgeneratoren gehen bei Hardware-Zufallsystemen die Gefahren zum größten Teil von der Unzuverlässigkeit sowie Beeinflussbarkeit der Hardware aus. Das Resultat bei einer wie auch immer gearteten Fehlfunktion ist stets eine Verringerung der „Zufälligkeit“ der Ausgabe, bis hin zu deren absoluter Vorhersagbarkeit. Im Folgenden werden die möglichen Fehlerquellen sowie Möglichkeiten, diese zu beseitigen oder zumindest zu erkennen, genannt.

**5.1. Versagen der Hardware.** Die Elemente, aus denen das Zufall erzeugende System zusammengesetzt ist, unterliegen nicht zu vermeidenden Fertigungsschwankungen sowie Verschleiß- und Alterungsprozessen und so kann es passieren, dass ein Baustein plötzlich ausfällt. Ein Ausfall in der Rechnersektion des Systems wird normalerweise schnell bemerkt. Kritischer sind solche Ausfälle, die nicht das ganze System betreffen, sondern lediglich die Verteilung der Ausgabe.

Um Hardwarefehler zu erkennen, muss ein Zufall erzeugendes System die Möglichkeit eines Selbst-Tests besitzen, der eventuelle Ausfälle aufdeckt [Fairfield *et al.* (1984)]. Zudem sollte die Ausgabe mittels statistischer Verfahren auf ihre Verteilung geprüft werden [Davis *et al.* (1994), Fairfield *et al.* (1984)].

**5.2. Beeinflussung von außen.** Alle physikalischen Effekte lassen sich durch äußere Einwirkung beeinflussen. Auch auf geschlossene Systeme ist eine Einflussnahme mehr oder weniger gut möglich, zum Beispiel durch Temperaturänderung oder elektromagnetische Felder. Die Beeinflussung kann bewusst von einem Angreifer vorgenommen werden oder einfach durch die äußeren Umstände des Systems begründet sein. Durch die Manipulation des physikalischen Effektes kann es passieren, dass dieser seine Unvorhersagbarkeit verliert.

Äußere Einflüsse müssen schon bei der Wahl der Zufallsquelle mit bedacht werden. Das Zufall erzeugende System muss in der Lage sein, sämtliche Einflüsse auf den zu Grunde liegenden physikalischen Effekt zu ignorieren; diese Eigenschaft wird auch als *Common Mode Rejection* bezeichnet [Jay (1984)]. Natürlich müssen nur solche Einflüsse in Betracht gezogen werden, die realistischerweise auf das System einwirken können. (Beispielsweise muss das System nicht robust gegenüber Hitze über 200°C sein, da bei solchen Temperaturen ohnehin die Rechneinheit ausfallen wird.)

**5.3. Unzureichende Entzerrung.** Wenn zur Entzerrung ein Verfahren angewandt wird, das auf eine bestimmte maximale Abweichung von der Gleichverteilung abgestimmt ist, weicht bei Überschreitung dieser Abweichung automatisch auch die Ausgabe der Entzerrungsfunktion weiter von der Gleichverteilung ab als beim Systementwurf für zulässig erachtet.

Diese Fehlfunktion lässt sich dadurch vermeiden, dass für die gewählte Entzerrungsmethode begründete, möglichst pessimistische Annahmen bezüglich der Verteilung des rohen Zufalls gemacht werden. Solche Annahmen sollten sich stets auf reproduzierbare Ergebnisse aus Versuchen mit dem zu Grunde liegenden physikalischen Effekt stützen.

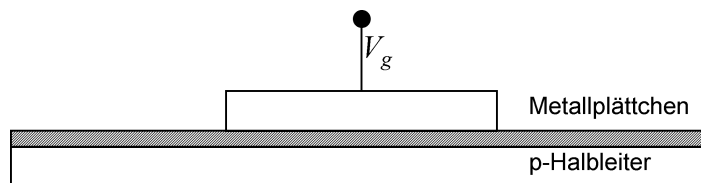
## 6. Beispiele für Zufall erzeugende Systeme

Im Folgenden werden drei unterschiedliche Ansätze vorgestellt, Zufall erzeugende Systeme zu verwirklichen. Die Beispiele unterscheiden sich sowohl in den physikalischen Zufallsquellen als auch in ihrer Komplexität.

### 6.1. Kriechströme in Halbleitern.

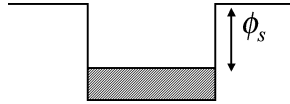
**6.1.1. Grundlage.** Benutzt wird ein Metallisolatorhalbleiterkondensator (engl. metal insulator semiconductor capacitor, MISC). Dieses VLSI-Element (engl. very large scale integration) findet sich aufgrund seiner Lichtempfindlichkeit normalerweise in großer Anzahl als sogenanntes CCD (engl. charge coupled device) in elektronischen Kameras. Zur Zufallsgewinnung ist jedoch eine andere Eigenschaft des MISC von Interesse: Lädt man den Kondensator auf, entstehen in einem zufälligen Prozess Kriechströme und erzeugen ein Ladungspotenzial, das gemessen werden kann.

**6.1.2. Verfahren.** Der MISC besteht aus einem positiv dotierten Halbleitersubstrat, einer darauf aufgetragenen dünnen Isolatorschicht aus Siliziumoxid ( $\text{SiO}_2$ ) und darauf einem Metallkontakt mit angeschlossenem Leiter (siehe Figur 6.1).



Figur 6.1: Aufbau eines MISC

Im Halbleitersubstrat herrscht aufgrund der positiven Dotierung ein Mangel an freien Elektronen. Wenn nun an dem Metallplättchen eine positive Spannung  $V_g$  angelegt wird, entsteht im Substrat unterhalb des Plättchens eine Potenziensenke, also ein Anziehungspunkt für Elektronen, die zu der positiven Ladung streben. Da im Substrat Elektronenmangel herrscht, kann die Senke nicht vollständig gefüllt werden und eine Elektronenknappheit entsteht. Die Tiefe der Senke wird durch  $V_g$  bestimmt, die benötigte Ladung (also die Anzahl an Elektronen) zum Füllen der Senke durch  $\varphi_s$  (siehe Figur 6.2). Dieses Ungleichgewicht wird schließlich dadurch aufgehoben, dass Elektronen erzeugt werden und die Senke auffüllen. Freie Elektronen werden entweder durch Lichteinfall auf das Halbleitersubstrat oder durch temperaturabhängige Kriechströme erzeugt. Um den Lichteffekt auszuschließen, wird der Kondensator gegen Lichteinfall abgeschirmt.



Figur 6.2: Potenziensenke im Halbleitersubstrat

Nach einer festgelegten Zeit, der sogenannten Integrationsperiode, wird die Spannung  $V_g$  wieder weggenommen und gleichzeitig die Spannung am Metallplättchen (und damit die Anzahl der bis dahin angesammelten Elektronen) über einen Messverstärker gemessen. Die Anzahl der durch Kriechströme erzeugten Elektronen folgt nach allgemein übereinstimmender Auffassung einem sogenannten Poisson-Prozess, das heißt die Varianz entspricht dem Erwartungswert [Hobson (1978)]. Zudem zeigt sich, dass die Kriechstromerzeugung zweier solcher Kondensatoren, auch in geringem räumlichen Abstand zueinander, nicht korreliert ist. Diese Eigenschaft führt zur Implementierung eines Zufall erzeugenden Systems:

Man positioniert zwei MISCs in geringem räumlichen Abstand, lädt sie über den gleichen Zeitraum mit der gleichen Spannung auf, und misst schließlich an beiden die entstandene Spannung. Der Differenz zwischen den Spannungen wird schließlich entweder eine „0“ oder eine „1“ zugewiesen, je nachdem, ob sie positiv oder negativ ist. Alle äußeren Einflüsse werden so eliminiert, da diese sich auf Grund der räumlichen Nähe gleichermaßen auf beide Kondensatoren auswirken und bei der Differenzbildung wegfallen.

Da die Menge der erzeugten Elektronen stark von der Temperatur abhängig ist (siehe Tafel 6.1) und die Genauigkeit heutiger Messverstärker zirka 100 Elektronen beträgt, muss die Integrationsperiode entsprechend lang gewählt werden, um die Zufälligkeit der Ausgabe sicher zu stellen. Tafel 6.2 zeigt die

Temperatur	erzeugter Kriechstrom
20°C	100 pA
-10°C	10 pA
-30°C	1 pA

Tafel 6.1: Temperaturabhängigkeit

Temperatur	$e = 0,005$	$e = 0,025$	$e = 0,05$
20°C	10 ms	300 $\mu$ s	60 $\mu$ s
-10°C	100 ms	3 ms	600 $\mu$ s
-30°C	1 s	30 ms	6 ms

Tafel 6.2: benötigte Integrationszeiten

jeweils notwendigen Integrationszeiten für verschiedene Temperaturen und gewünschte maximale Abweichungen von der Gleichverteilung (siehe Kapitel 4.2).

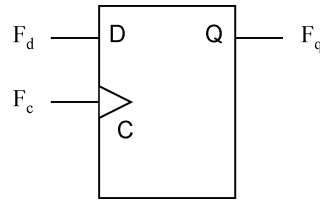
**6.1.3. Zusammenfassung.** Die Zufallserzeugung über Kriechströme in Halbleitern hat den Vorteil der Realisierbarkeit in VLSI-Schaltkreisen. Es liegen jedoch noch keine statistischen Tests über die Güte des erzeugten Zufalls vor. Mit diesem Verfahren lassen sich in der Größenordnung von 1000 zufälligen Bits pro Minute erzeugen.

## 6.2. Periodenschwankungen bei Oszillatoren.

**6.2.1. Grundlage.** Elektronische Oszillatoren besitzen stets ein gewisses Maß an Frequenzinstabilität. Diese Ungleichmäßigkeiten sind am geringsten bei Kristalloszillatoren und am größten bei RC-Oszillatoren (Widerstand-Kondensator-Schaltung) [Abidi & Meyer (1983)]. Da der Zufall aus den Periodenschwankungen gewonnen werden soll, benutzt man hier einen RC-Oszillator. Die Taktflanken dieses ungenauen Oszillators bestimmen die Zeitpunkte, an denen der Ausgangswert eines anderen, schnelleren Oszillators (nicht notwendigerweise RC-Typ) gemessen wird. Diese unregelmäßig gemessenen Bits sind der rohe Zufall. Um gefährliche (weil zufallsmindernde) Koppelung zu vermeiden, ist es dabei wichtig, dass die beiden Oszillatoren nicht auf dem selben Chip sitzen.

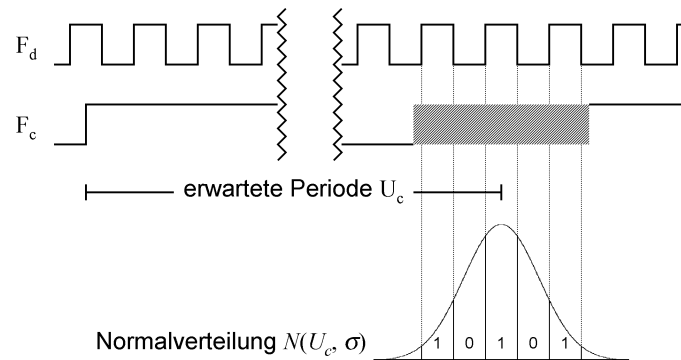
**6.2.2. Verfahren.** Durchgeführt wird die Messung mittels eines D-Flipflops (siehe Figur 6.3). Am Ausgang  $F_q$  des D-Flipflops liegt dabei immer der zuletzt gemessene Wert vom Eingang  $F_d$  (verbunden mit dem Ausgang

des hochfrequenten Oszillators) an, bis bei steigender Flanke an  $F_c$  (verbunden mit dem ungenauen RC-Oszillator) erneut  $F_d$  gemessen bzw. gespeichert wird.



Figur 6.3: D-Flipflop

Da die Periodenlänge des RC-Oszillators durch eine Normalverteilung  $N(U_c, \sigma)$  modellierbar ist, kann man leicht die Anforderung an die Ungenauigkeit des Oszillators formulieren: Die Standardabweichung  $\sigma$  muss mindestens so groß sein wie die Periodenlänge  $T_d$  des gemessenen hochfrequenten Oszillators, damit aufeinanderfolgende gemessene Bits nicht vorhersagbar werden (siehe Figur 6.4).



Figur 6.4: Verteilung der Periodenlänge des RC-Oszillators

Da Oszillatoren nicht beliebig ungenau hergestellt werden können, benutzt man eine Eigenschaft der Normalverteilung, um dieser Anforderung gerecht zu werden: ändert man die Zufallsvariable, so ändert sich proportional auch die Standardabweichung. Es werden die gemessenen Bits derart gemischt, dass die letztlich ausgegebenen Bits aus Messungen stammen, die mindestens 429 Taktzyklen auseinander lagen. Damit erhöht sich auch die Standardabweichung um das 429-fache; das heißt, die Periodenstandardabweichung des RC-Oszillators muss nur noch mindestens  $\frac{1}{429}$  von  $T_d$  betragen.

Verwirklicht wird das Mischen der Bits durch einen sogenannten Scrambler-Schaltkreis (engl. mischen, verschlüsseln), der jeweils 2680 gemessene Bits in ein 536 Bit breites Schieberegister mischt. Dabei wird gleichzeitig durch Exklusiv-Oder-Verknüpfungen (also Parität) von jeweils 5 Bits eine Entzerrung vorgenommen.

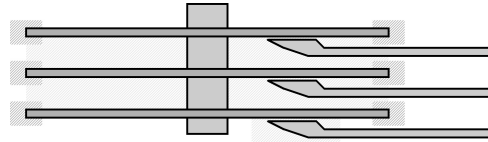
**6.2.3. Selbsttest.** Bevor die erzeugten Zufallsbits aus dem Schieberegister ausgelesen werden können, wird zunächst ein statistischer Test durchgeführt, der als Selbsttest des Systems fungiert. Das Register wird 1000 Mal mit Zufallsbits gefüllt und jedes Mal werden die letzten 4 Bits des Registers mit einem beliebigen aber festen 4-Bit-Muster verglichen, bei Übereinstimmung wird ein Zähler erhöht. Der Erwartungswert für den Zähler beträgt nach 1000 Durchläufen  $1000 \cdot \frac{1}{2^4} = 62,5$ , die Standardabweichung beträgt  $7,65$ . Der tatsächliche Wert des Zählers wird schließlich mit einer frei wählbaren Unter- sowie Obergrenze verglichen. Falls der Wert außerhalb der gewählten Grenzen liegt, springt das System in einen Alarm-Zustand und verweigert das Auslesen des Schieberegisters. Dies bedeutet, dass zum Beispiel bei einer gewählten Untergrenze von  $62,5 - 2 \cdot 7,65$  (Erwartungswert  $-2 \cdot$  Standardabweichung) und einer gewählten Obergrenze von  $62,5 + 2 \cdot 7,65$  (Erwartungswert  $+2 \cdot$  Standardabweichung) auch bei einer perfekten Gleichverteilung der erzeugten Bits im Durchschnitt jeder zwanzigste Test fehlschlägt. Dies lässt sich vermeiden, indem man das Intervall zum Bestehen des Tests entsprechend vergrößert.

**6.2.4. Zusammenfassung.** Die Zufallserzeugung über Periodenschwankungen von Oszillatoren ist bereits von Fairfield *et al.* (1984) in einem LSI-Schaltkreis realisiert worden. Vorteilhaft ist die eingebaute Selbsttest-Funktion. Die Güte des erzeugten Zufalls hängt jedoch stark vom jeweils benutzten Oszillator ab. Bei einer mittleren Frequenz von 1 kHz des RC-Oszillators lassen sich mit diesem Verfahren 1608 zufällige Bits pro Minute erzeugen.

### 6.3. Luftturbulenzen bei Festplatten.

**6.3.1. Grundlage.** Eine Festplatte besteht im Wesentlichen aus mehreren Scheiben, die um eine gemeinsame Achse rotieren, sowie mehreren miteinander verbundenen Schreib-/Lesearmen, die zwischen der Scheibenmitte und dem Scheibenrand hin- und hergeschwenkt werden. Umschlossen wird das Ganze von einem zylindrischen Gehäuse. An verschiedenen Stellen dieses geschlossenen Systems treten Luftturbulenzen auf (siehe Figur 6.5).

**6.3.2. Verfahren.** Der Luftstrom am Rande der einzelnen Scheiben (genannt Taylor-Couette-Strom) bringt schon nach einer kurzen Zeit der Umdrehung vollkommen zufällige Turbulenzen hervor, die erwiesenermaßen die



Figur 6.5: Luftturbulenzen in einem Festplattensystem

Umdrehungsgeschwindigkeit der Scheiben beeinflussen. Diese Beeinflussung ist zwar nur minimal, aber messbar. Gemessen werden tatsächlich die Zugriffszeiten für einen bestimmten Datenblock auf der Festplatte. Da diese Zugriffszeiten jedoch stets stark strukturiert und korreliert sind, ist eine starke Entzerrung notwendig. Zu diesem Zweck verwendet man hier die schnelle Fourier-Transformation (FFT) nach Brillinger (1981), mittels derer die Zugriffszeiten in Frequenzspektren überführt werden, deren vorhersagbare Spitzen daraufhin entfernt werden. Dies führt zu unabhängigen, normalverteilten Spektren, die somit echten Zufall darstellen.

**6.3.3. Zusammenfassung.** Die Zufallserzeugung über Luftturbulenzen in Festplatten ist relativ leicht möglich, da in den meisten Computern existierende Hardware (eben die Festplatte) genutzt werden kann und lediglich softwareseitig die Entzerrung vorgenommen werden muss. Zudem sorgen die Sicherheitsmechanismen des Betriebssystems dafür, dass Hardwarefehler erkannt werden. Die Anwendung beschränkt sich jedoch auf heutige Rechner mit Plattenlaufwerken, ausgeschlossen bleiben zum Beispiel Handheld-Computer mit statischem Speicher. Mit diesem Verfahren lassen sich etwa 100 zufällige Bits pro Minute erzeugen.

## 7. Fazit, Ausblick

Echter Zufall ist in der Kryptographie unabdingbar, er kann durch Pseudozufall schnell und sicher vervielfältigt, jedoch niemals ersetzt werden. Aus vielen physikalischen Effekten lässt sich echter Zufall gewinnen; es muss jedoch darauf geachtet werden, dass der zugrunde liegende Effekt nicht hinreichend detailliert modelliert werden kann (dies kann jedoch immer nur eine durch Beobachtungen gefestigte Annahme sein, da sich ein physikalischer Effekt letztlich nicht beweisen lässt). Ebenso müssen Maßnahmen getroffen werden, die eine bestimmte Güte des erzeugten Zufalls garantieren sowie eine äußere Einflussnahme verhindern. Die vorgestellten Beispiele zeigen, dass schon bald ohne größeren Aufwand in jedem Computer eine Möglichkeit zur Gewinnung echten Zufalls vorhanden sein könnte.

---

## Literatur

ASAD A. ABIDI & ROBERT G. MEYER (1983). Noise in Relaxation Oscillators. *IEEE Journal of Solid-State Circuits* **SC-18**(6), 794–802.

G. B. AGNEW (1987). Random Sources for Cryptographic Systems. In *Advances in Cryptology: Proceedings of EUROCRYPT 1987*, Amsterdam, The Netherlands, DAVID CHAUM & WYN L. PRICE, editors, number 304 in Lecture Notes in Computer Science, 77–81. Springer-Verlag. ISBN 3-540-19102-X. ISSN 0302-9743.

DAVID R. BRILLINGER (1981). *Time Series: Data Analysis and Theory*. Holden-Day, San Francisco, CA.

DON DAVIS, ROSS IHAKA & PHILIP FENSTERMACHER (1994). Cryptographic randomness from air turbulence in disk drives. In *Advances in Cryptology: Proceedings of CRYPTO '94*, Santa Barbara CA, YVO G. DESMEDT, editor, number 839 in Lecture Notes in Computer Science, 114–120. Springer-Verlag, Santa Barbara, CA, USA.

DONALD E. EASTLAKE, STEPHEN D. CROCKER & JEFFREY I. SCHILLER (1994). Randomness Recommendations for Security. URL <http://www.faqs.org/rfcs/rfc1750.html>.

R. C. FAIRFIELD, R. L. MORTENSON & K. B. COULTHART (1984). An LSI random number generator (RNG). In *Advances in Cryptology: Proceedings of CRYPTO '84*, Santa Barbara CA, G. R. BLAKLEY & DAVID CHAUM, editors, number 196 in Lecture Notes in Computer Science, 203–230. Springer-Verlag. ISBN 3-540-15658-5. ISSN 0302-9743.

G. HOBSON (1978). *Charge-Transfer Devices*. Edward Arnold, London.

F. JAY (editor) (1984). *IEEE Standard Dictionary of Electrical and Electronic Terms*. IEEE Press, New York.

JOHN VON NEUMANN (1963). Various Techniques Used in Connection With Random Digits. In *John von Neumann, Collected Works*, volume 5, 768–770. Pergamon Press, Oxford.

CLAUDE E. SHANNON (1948). A Mathematical Theory of Communication. *Bell System Technical Journal* **27**, July (pp. 379–423), October (pp. 623–656).

KRISTIAN WILKENING

Pohlweg 60a

33098 Paderborn

kristian@upb.de