Gottfried Wilhelm Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Theoretische Informatik

Statistische Tests in der Kryptographie

Bachelorarbeit

Julian Müller
Matrikelnummer: 10004829

Erstprüfer: Prof. Dr. Heribert Vollmer
Zweitprüfer: Dr. Arne Meier

4. Mai 2020

Inhaltsverzeichnis

1 Einleitung

Informationssicherheit betrifft Menschen auf der ganzen Welt. Sei es beim Surfen im Internet, beim Schicken von Emails, oder beim Onlinebanking. Überall müssen Informationen sicher verwahrt oder von A nach B gebracht werden und trotzdem jederzeit auf Abruf erreichbar sein. Dies verlangt nach schnellen und schwer zu brechenden Sicherheitsmechanismen. In der Kryptographie werden dafür oft Zufallszahlen verwendet. Diese bieten den Vorteil, dass sie von Angreifern nur schwer bis gar nicht reproduziert werden können, wodurch ein Verfahren, das Zufallszahlen nutzt, meistens ein großes Maß an Sicherheit bietet.

Solche Zufallszahlen können jedoch nicht unmittelbar von Computern generiert werden, da diese deterministisch arbeiten. Daher greift man auf physikalische Vorgänge zurück und nutzt deren echt zufällige Ausgaben. Beispiele sind komplexe Systeme wie radioaktive Zerfälle, das Rauschen eines ohmschen Widerstands, aber auch das simple Werfen einer Münze. Während die echte Zufälligkeit ein großer Vorteil dieser Prozesse ist, stellt ihre Ineffizienz ein Problem für die praktische Anwendung dar. Beim Verschlüsseln von Nachrichten werden oft pro Nachricht viele Zufallszahlen benötigt, um eine sichere Übertragung zu gewährleisten. Physikalische Vorgänge generieren für diese und weitere Anwendungen zu wenig Ausgaben in einer gewissen Zeit. Um dennoch zufällige Zahlen nutzen zu können, gibt es sogenannte Pseudozufallszahlengeneratoren.

Dies sind effiziente Algorithmen, die aus der kurzen, echt zufälligen Ausgabe eines physikalischen Vorgangs eine deutlich längere Zahlenfolge bilden. Diese längeren Zahlenfolgen nennt man Pseudozufallszahlen, da sie nicht durch echten Zufall entstanden sind, aber dennoch Eigenschaften von echt zufälligen Zahlen besitzen. Entsprechend sind sie einerseits effizient zu generieren und bieten andererseits trotzdem die nötige Sicherheit, die in heutigen kryptographischen Anwendungen benötigt wird. Diese Sicherheit basiert auf der Tatsache, dass eine Pseudozufallszahl nicht von einer echten Zufallszahl unterschieden werden kann. Einem Angreifer ist es daher nicht möglich ohne Weiteres Aussagen über eine solche Zahl zu treffen. Probleme treten jedoch dann auf, wenn der Algorithmus zur Generierung dieser Zahlen kein Pseudozufallszahlengenerator ist. Das führt etwa dazu, dass Vorhersagen über Ausgaben des Algorithmus getroffen werden können. Um dem entgegenzuwirken, prüft man diese Ausgaben auf Pseudozufälligkeit. Hierzu verwendet man statistische Tests.

Ein statistischer Test ist formal betrachtet eine Funktion, die aus einer Beobachtung oder Stichprobe auf die Gültigkeit einer Hypothese abbildet. Man spricht auch von einer Entscheidung für bzw. gegen eine Nullhypothese H_0. Im Kontext dieser Arbeit schreibt die Aussage der Nullhypothese den zu testenden Zahlenfol-

gen eine Zufälligkeit zu. Wir betrachten alle Zahlenfolgen als Binärzahlen. Sei H_0 etwa die Aussage, dass eine Zahlenfolge ungefähr die gleiche Anzahl Nullen und Einsen besitzt. Ein Test, der anhand einer Zahlenfolge eine Entscheidung über die Gültigkeit dieser Nullhypothese treffen soll, vergleicht also die Anzahl vorkommender Nullen und Einsen miteinander und trifft dann anhand der beiden ermittelten Werte eine Entscheidung. Bei einer großen Differenz der Werte lehnt der Test H_0 ab, bei kleiner Differenz bestätigt er H_0. Statistische Tests liefern jedoch keine unfehlbaren Entscheidungen. Die Stichprobe, auf der ein Test sein Ergebnis begründet, unterliegt zufälligen Vorgängen. Darum ist es für Tests nicht möglich eine Entscheidung mit vollkommener Sicherheit zu treffen. Stattdessen arbeitet man mit Fehlerwahrscheinlichkeiten um dem Testergebnis eine gewisse Signifikanz zuschreiben zu können.

In Kapitel 2 wird genauer auf die mathematischen Konzepte hinter statistischen Tests eingegangen. Benötigte stochastischen Grundlagen werden vermittelt und grundlegende Begriffe der Kryptographie definiert. Kapitel 3 stellt Anwendungsbereiche von statistischen Tests und Zufallszahlen vor und beschreibt die Ausgangslage für das Testen einer Zahlenfolge auf Zufälligkeit. Ein verbreitetes Testset, die sogenannte NIST Statistical Test Suite, wird in Kapitel 4 vorgestellt und ausgewählte Tests werden detailliert beschrieben. In Kapitel 5 wird schlussendlich das statistische Testen eines Zufallszahlengenerators anhand des Micali-Schnorr Generators durchgeführt.

2 Grundlagen

2.1 Grundlagen der Stochastik

Im Folgenden werden einige mathematische Begriffe und Grundlagen der Stochastik definiert, welche in dieser Arbeit genutzt werden. Als Orientierung dient hierfür das Buch *Stochastik für Einsteiger* von Henze[6].

Definition 2.1.1 (Zufallsvariable)**.** Ist Ω eine nichtleere Menge, so heißt jede Abbildung

$$X\colon \Omega \to \mathbb{R}$$

von Ω in die Menge $\mathbb{R}$ der reellen Zahlen *Zufallsvariable* (auf Ω). Ω nennt man die *Ergebnismenge*. Statt $X(\omega)$ mit $\omega \in \Omega$ wird häufig nur X geschrieben. Enthält Ω endlich viele oder abzählbar unendlich viele Elemente, heißt X *diskrete Zufallsvariable*.

Definition 2.1.2 (Potenzmenge)**.** Sei M eine Menge. Die *Potenzmenge* von M, geschrieben $\mathfrak{P}(M)$, ist eine Menge, die jede Teilmenge von A enthält:

$$\mathfrak{P}(M) = \{N \mid N \subseteq M\}.$$

Im Mittelpunkt der Stochastik steht das *Zufallsexperiment*. Es bezeichnet einen Versuch, dessen Durchführung unter genau festgelegten Rahmenbedingungen zu einem von mehreren bekannten *Ergebnissen* führt. Ein Beispiel ist das Werfen eines sechsseitigen Spielwürfels. Beim Werfen können als Ergebnisse die Augenzahlen 1, 2, 3, 4, 5 und 6 eintreten. Diese stellen wir formal als ihren jeweiligen Zahlenwert dar. Wir bilden also in diesem Fall das Ergebnis „Augenzahl“ auf eine reelle Zahl ab und können daher von einer Zufallsvariablen sprechen. Welches Ergebnis eintritt ist vor der Durchführung ungewiss. Diese Ungewissheit beschreibt man mit *Wahrscheinlichkeiten*. Formal wird ein Zufallsexperiment und dessen Rahmenbedingungen mit einem sogenannten *Wahrscheinlichkeitsraum* dargestellt.

Definition 2.1.3 (Wahrscheinlichkeitsraum)**.** Ein *Wahrscheinlichkeitsraum* (kurz: *W-Raum*) ist ein Paar (Ω, P), wobei Ω eine nichtleere Menge und P eine Funktion mit $P\colon \mathfrak{P}(\Omega) \to [0,1]$ ist und folgende Eigenschaften besitzt:

a) $P(A) \geq 0$ für alle $A \subseteq \Omega$, (Nichtnegativität)

b) $P(\Omega) = 1$, (Normiertheit)

c) $P(A+B) = P(A) + P(B)$, falls $A \cap B = \emptyset$. (Additivität)

P heißt *Wahrscheinlichkeitsmaß* (oder auch *Wahrscheinlichkeitsverteilung*) auf der Ergebnismenge Ω, bzw. auf den Teilmengen von Ω. Jedes $\omega \in \Omega$ heißt *Ergebnis* und jedes $A \subseteq \Omega$ ein *Ereignis* des zugrundeliegenden Zufallsexperiments. Bei Ereignissen, die nur ein einziges Ergebnis enthalten spricht man von einem *Elementarereignis* $\{\omega\}$. Die Zahl $P(A)$ heißt *Wahrscheinlichkeit* des Ereignisses A.

Alternativ kann ein Wahrscheinlichkeitsraum auch mit einer Zufallsvariablen statt einer Ergebnismenge definiert werden, wobei hier der Wertebereich der Zufallsvariablen als Ergebnismenge des Wahrscheinlichkeitsraums gewählt wird. Das Wahrscheinlichkeitsmaß bezeichnet man dann auch als *Verteilung der Zufallsvariablen.*

Beispiel 2.1 (Zweimaliger Würfelwurf)**.** Für dieses und alle folgenden Beispiele verwenden wir einen fairen, sechsseitigen Würfel.
Wir werfen einen Würfel zweimal hintereinander und notieren uns die beiden geworfenen Augenzahlen. Dieser Versuch kann als Zufallsexperiment bezeichnet werden und somit kann ein entsprechender Wahrscheinlichkeitsraum (Ω, P) definiert werden. Die Ergebnisse des Zufallsexperiments sind Paare von Zahlen von Eins bis Sechs. Folglich definieren wir die Ergebnismenge des Wahrscheinlichkeitsraums mit $\Omega = \{(\omega_1, \omega_2) \mid \omega_1, \omega_2 \in \{1, 2, 3, 4, 5, 6\}\}$. Da es sich um einen fairen Würfel handelt, besitzen alle Ergebnisse ω, bzw. alle Elementarereignisse $\{\omega\}$ die gleiche Wahrscheinlichkeit. Aufgrund der Normiertheit des Wahrscheinlichkeitsmaßes und weil es insgesamt 36 Ergebnisse gibt muss gelten: $P(\{\omega\}) = \frac{1}{36}$. Für alle Ereignisse $A \subseteq \Omega$ gilt entsprechend $P(A) = \frac{|A|}{36}$. Diese Wahrscheinlichkeitsverteilung P nennen wir *Gleichverteilung.*

Definition 2.1.4 ((Diskrete) Gleichverteilung)**.** Sei Z eine diskrete Zufallsvariable und (Z, P) ein Wahrscheinlichkeitsraum. Man sagt P ist die *Gleichverteilung* von Z oder Z *ist gleichverteilt*, wenn für alle $z \in Z$ und alle $A \in \mathfrak{P}(Z)$ gilt:

$$P(z) = \frac{1}{|Z|} \text{ und } P(A) = \frac{|A|}{|Z|}.$$

Definition 2.1.5 (Erwartungswert (einer diskreten Zufallsvariable))**.** Für eine Zufallsvariable $X \colon \Omega \to \mathbb{R}$ auf einem W-Raum $(\Omega,\ P)$ heißt

$$E(X) \coloneqq \sum_{\omega \in \Omega} X(\omega) P(\omega)$$

der Erwartungswert von X.

Definition 2.1.6 (Varianz und Standardabweichung)**.** Für eine Zufallsvariable $X\colon \Omega \to \mathbb{R}$ auf einem W-Raum (Ω, P) heißt

$$V(X) \coloneqq E((X - E(X))^2)$$

die *Varianz* von X. Alternativ bezeichnet man $V(X)$ auch als $\sigma^2(X)$ oder σ_X^2. Die (positive) Wurzel

$$\sigma(X) \coloneqq \left|\sqrt{V(X)}\right|$$

aus $V(X)$ heißt *Standardabweichung* von X.

Definition 2.1.7 (Standardisierung einer Zufallsvariablen)**.** Sei X eine Zufallsvariable, so heißt die Transformation

$$X \to X^* \coloneqq \frac{X - E(X)}{\sigma(X)}$$

die *Standardisierung* von X. Eine sogenannte *standardisierte* Zufallsvariable X^* besitzt einen Erwartungswert von $E(X^*) = 0$ und eine Varianz von $V(X^*) = 1$.

Definition 2.1.8 (Stochastische Unabhängigkeit)**.** Seien (Ω, P) ein W-Raum und $A_1, ..., A_n$ Ereignisse $(n \geq 2)$. $A_1, ..., A_n$ heißen *(stochastisch) unabhängig*, genau dann wenn gilt:

$$P\left(\bigcap_{j\in J} A_j\right) = \prod_{j\in J} P(A_j)$$

für jede mindestens zweielementige Menge $J \subseteq \{1, 2, ..., n\}$.

2.2 Einführung in statistische Tests

Wenn in einem Wahrscheinlichkeitsraum das W-Maß unbekannt ist, werden mehrere W-Maße als möglich angenommen. Für einen solchen Fall definieren wir ein sogenanntes *statistisches Modell*.

Definition 2.2.1 (Statistisches Modell)**.** Ein *statistisches Modell* ist ein Paar $(\Omega, (P_\vartheta)_{\vartheta\in\Theta})$, wobei für ein unbekanntes $\vartheta \in \Theta$ angenommen wird, dass der zugehörige W-Raum (Ω, P_ϑ) eines Zufallsexperiments vorliegt. Das entsprechende ϑ heißt *wahrer* Parameter. Das dadurch bestimmte W-Maß P_ϑ ist die *wahre Verteilung* von Ω. Auch hier kann analog zum Wahrscheinlichkeitsraum eine Zufallsvariable zur Definition der Ergebnismenge genutzt werden.

Bevor der Begriff des Tests genauer definiert wird, wollen wir den Kontext in welchem dieser genutzt wird näher erläutern. Einem Test geht immer ein sogenanntes Testproblem voraus, für welches eine Lösung gefunden werden soll.

Definition 2.2.2 (Testproblem)**.** Sei $M = (\Omega, (P_\vartheta)_{\vartheta\in\Theta})$ ein statistisches Modell. Für ein Testproblem (M, Θ_0, Θ_1) zerlegt man die Menge Θ in zwei disjunkte nichtleere Teilmengen Θ_0 und Θ_1. Es gilt also: $\Theta = \Theta_0 \uplus \Theta_1$. Das Ziel eines Tests ist es eine Entscheidung zwischen

$$H_0 : \vartheta \in \Theta_0 \text{ und } H_1 : \vartheta \in \Theta_1$$

zu treffen. Man nennt H_0 die *Nullhypothese* und H_1 die *Gegenhypothese*. Ein Testproblem formuliert also die Zerlegung von Θ und gibt somit die beiden Hypothesen vor zwischen welchen ein Test entscheiden muss. Wir sagen auch: *Der Test entscheidet das Testproblem*. Zur Definition von H_0 und H_1 nutzt man oft einen Parameter ϑ_0, welcher in Θ_0 enthalten ist und die „Grenze“ zu Θ_1 darstellt. Θ_0 ist meistens von der Form $\Theta_0 = \{\vartheta \mid \vartheta \leq \vartheta_0\}$, wobei diese analog auch mit anderen Vergleichsoperatoren dargestellt werden kann.

Um sinnvolle Lösungen eines Testproblems zu erhalten muss das Festlegen von H_0 und H_1 aus dem Sachkontext des zugrundeliegenden Zufallsexperiments heraus erfolgen. Dies wird im folgenden Beispiel 2.2 genauer erläutert.

Beispiel 2.2 (Unfairer Würfel)**.** Wir vermuten, dass ein vorliegender Würfel manipuliert wurde, sodass er häufiger eine Sechs würfelt als ein fairer Würfel. Wir definieren eine Zufallsvariable $Z\colon \{1,2,3,4,5,6\} \to \{0,1\}$ mit:

$$Z(\omega) = \begin{cases} 1 & \text{, falls eine Sechs gewürfelt wird} \\ 0 & \text{, sonst} \end{cases}$$

und ein entsprechendes stochastisches Modell mit $M = (Z, (P_\vartheta)_{\vartheta\in\Theta})$. Wir suchen die Wahrscheinlichkeit eine Sechs zu würfeln $\vartheta = P(Z = 1)$. Entsprechend muss $P(Z = 0) = 1 - \vartheta$ sein. Formal würde beim Zutreffen unserer Vermutung für die Wahrscheinlichkeit ϑ gelten: $\vartheta > \frac{1}{6}$, bei falscher Vermutung $\vartheta \leq \frac{1}{6}$. Aus diesem Sachkontext heraus können wir nun sinnvolle Hypothesen für das Testproblem aufstellen. Das zugehörige Testproblem $L = (M, \Theta_0, \Theta_1)$ zerlegt die Menge Θ in $\Theta_0 = \left\{\vartheta \mid 0 \leq \vartheta \leq \frac{1}{6}\right\}$ und $\Theta_1 = \left\{\vartheta \mid \frac{1}{6} < \vartheta \leq 1\right\}$. Ein Test auf diesem Testproblem müsste also zwischen $H_0 : \vartheta \in \Theta_0$ und $H_1 : \vartheta \in \Theta_1$ entscheiden. Zu Erwähnen ist hier, dass dieses Testproblem lediglich die Frage danach stellt, ob die Sechs häufiger gewürfelt wird als bei einem fairen Würfel. Ob die Sechs eventuell sogar seltener gewürfelt wird als in einem aus sechs Fällen, ist für einen Test auf diesem Testproblem unmöglich zu entscheiden. Im Folgenden sprechen wir nur dann von einem unfairen Würfel, wenn die Sechs häufiger als bei einem fairen Würfel fällt.

Im vorangehenden Beispiel 2.2 beschreibt die Zufallsvariable Z eine Zufallsvariable, die entweder den Wert 1 oder den Wert 0 annehmen kann. In einem solchen Fall nennt man die Werte des Wertebereichs $\{0,1\}$ *Treffer* $(Z = 1)$ und *Nieten* $(Z = 0)$. Außerdem bezeichnet man Z als *bernoulli-verteilt*.

Definition 2.2.3 (Bernoulli-Verteilung)**.** Eine Zufallsvariable $Z : \Omega \to \{0,1\}$ besitzt eine Bernoulli-Verteilung mit *Trefferwahrscheinlichkeit* p, kurz: $Z \sim \mathrm{Ber}(p)$, falls gilt:

$$P(Z = 1) = p.$$

Aufgrund der Normiertheit eines Wahrscheinlichkeitsmaßes folgt:

$$P(Z = 0) = 1 - p =: q.$$

Der Erwartungswert einer bernoulli-verteilten Zufallsvariable Z ist $E(Z) = p$. Die Varianz ist $V(Z) = pq$. Der Kürze halber schreiben wir q anstelle von $(1 - p)$.

Wenn das Testproblem formuliert ist, beginnt der eigentliche Test. Für einen Test benötigt man eine sogenannte *Stichprobe* S. Hierzu führt man das Zufallsexperiment n-mal durch und erhält so eine n-elementige Multimenge S von Ergebnissen. Wir sprechen von einer *Stichprobe vom Umfang* n.

Definition 2.2.4 (Statistischer Test)**.** Ein *statistischer Test* oder auch nur *Test* für das Testproblem $H_0 : \vartheta \in \Theta_0$ gegen $H_1 : \vartheta \in \Theta_1$ beinhaltet eine Testfunktion $f\colon \Omega^n \to \{0,1\}$ mit Stichprobe $S \in \Omega^n$ vom Umfang n, mit

$$f(S) = \begin{cases} 1 & \text{, falls } S \in K \\ 0 & \text{, falls } S \in \overline{K} \end{cases}$$

Die Menge $K \subsetneq \Omega^n$ heißt *kritischer Bereich*, die Menge $\overline{K} = \Omega^n \setminus K$ heißt *Annahmebereich* des Tests. Das *Testergebnis* ist die Bestätigung der Nullhypothese H_0 für $f(S) = 0$, oder das Ablehnen bzw. Verwerfen von H_0 für $f(S) = 1$.

In den folgenden Sachverhalten besitzt der kritische Bereich K immer eine Form der Art

$$K = \{S \in \Omega^n \mid T(S) \geq c\} =: \{T \geq c\},$$

wobei $T\colon \Omega^n \to \mathbb{R}$ eine Funktion und $c \in \mathbb{R}$ eine Konstante ist. T nennt man die *Teststatistik* und c den *kritischen Wert*. Die Nullhypothese H_0 wird also genau dann abgelehnt, wenn die Teststatistik mindestens den kritischen Wert c erreicht. Der kritische Bereich kann je nach Art des Testproblems auch durch $\{T < c\}$, $\{T > c\}$, $\{T \leq c\}$ oder mit anderen Vergleichsoperatoren beschrieben werden.

Beispiel 2.3 (Testen eines vermeintlich unfairen Würfels)**.** Wir gehen davon aus, dass das Testproblem L aus Beispiel 2.2 vorliegt. Wir möchten nun einen Test definieren, welcher unser Testproblem entscheidet. Zunächst müssen wir uns überlegen welchen Umfang n unsere Stichprobe besitzen soll. Der Einfachheit halber wählen wir $n = 120$. Wir haben also 120 unabhängige Ergebnisse ω_{s_i} des Würfelwurfs in unserer Stichprobe $S \in \Omega^{120}$ mit $1 \leq i \leq 120$. Jedes Ergebnis in der Stichprobe ist ein Wert der Zufallsvariable $Z \sim Ber\left(\frac{1}{6}\right)$. Der Erwartungswert von Z beträgt $E(Z) = \frac{1}{6}$, ausgehend davon, dass es sich um einen fairen Würfel handelt. Entsprechend kann man erwarten, dass $\frac{1}{6}$ der 120 Ergebnisse den Wert 1 annehmen. Dies entspräche 20 Ergebnissen. Sollten in unserer Stichprobe deutlich mehr Ergebnisse den Wert 1 annehmen, ließe dies darauf schließen, dass der Würfel häufiger eine Sechs würfelt als ein normaler Würfel und er wäre somit als unfair zu betrachten. Da es sich bei einem Würfelwurf jedoch nach wie vor um einen zufälligen Vorgang handelt, ist eine Abweichung von den 20 Ergebnissen nicht sofort als Beleg für ein Zutreffen der Gegenhypothese zu sehen. Wir überlegen uns ab wie vielen geworfenen Sechsen in unserer Stichprobe S (also Ergebnisse $\omega_{s_i} = 1$) wir uns dafür entscheiden würden, dass der Würfel zu häufig eine Sechs würfelt und somit unfair ist. Wir wählen diesen kritischen Wert mit $c = 24$. Die Anzahl an geworfenen Sechsen(Anzahl Treffer) definieren wir als die Teststatistik des Tests mit $T(S) = \sum_{i=1}^{n} \omega_{s_i}$. Da wir ab 24 gewürfelten Sechsen den Würfel als unfair betrachten, definieren wir den kritischen Bereich folgendermaßen: $K = \{S \mid T(S) \geq 24\} = \{T \geq 24\}$. Der Annahmebereich wäre $\overline{K} = \Omega^{120} \setminus K = \{S \mid T(S) < 24\} = \{T < 24\}$. Der Test auf dem Testproblem L lautet also letztendlich:

$$f(S) = \begin{cases} 1 & \text{, falls } S \in \{T \geq 24\} \\ 0 & \text{, falls } S \in \{T < 24\}. \end{cases}$$

Im vorangehenden Beispiel 2.3 betrachten wir für unsere Stichprobe die bernoulli-verteilte Zufallsvariable Z 120-mal. Betrachten wir nun die Anzahl der Treffer von Z, also in diesem Fall die Teststatistik T, erhalten wir eine neue Zufallsvariable $Y = \sum_{i=1}^{120} Z_i$, welche man *binomialverteilt* nennen.

Definition 2.2.5 (Binomialverteilung)**.** Eine Zufallsvariable Y besitzt eine Binomialverteilung mit Parametern n und p, kurz: $Y \sim \text{Bin}(n, p)$, falls gilt:

$$P(Y = k) = \binom{n}{k} p^k q^{n-k}, \text{ für alle } k \in \{0, 1, ..., n\}.$$

Hierbei beschreibt n die Anzahl der stochastisch unabhängig erzeugten Instanzen einer Bernoulli-verteilten Zufallsvariablen $Z \sim \text{Ber}(p)$. Wir nennen n die *Anzahl der Versuche*. Man sagt auch: Y *ist binomialverteilt*. Der Erwartungswert einer binomialverteilten Zufallsvariable Y ist $E(Y) = np$. Die Varianz ist $V(Z) = npq$. Auch hier schreiben wir q anstelle von $(1 - p)$. Es gilt:

$$P(Y \leq k) = \sum_{y=0}^{k} P(Y = y)$$

$$P(j \leq Y \leq k) = \sum_{y=j}^{k} P(Y = y), \text{ für alle } j \leq k \text{ mit } j, k \in \{0, 1, ..., n\}.$$

In unserem Beispiel 2.3 betrachtet unsere Teststatistik T also den Wert der Zufallsvariable $Y \sim Bin\left(120, \frac{1}{6}\right)$. Wir nennen $P(Y \leq k)$ die *Verteilungsfunktion von* Y.

Definition 2.2.6 (Verteilungsfunktion)**.** Ist X eine Zufallsvariable auf einem W-Raum (Ω, P), so heißt die durch

$$F(x) \coloneqq P(X \leq x), \text{ mit } x \in \mathbb{R},$$

definierte Funktion $F : \mathbb{R} \to [0, 1]$ die *Verteilungsfunktion* von X. Sie besitzt folgende Eigenschaften:

a) F ist *monoton wachsend*, d.h., aus $x \leq y$ folgt stets $F(x) \leq F(y)$.

b) F ist *rechtsseitig stetig*, d.h., es gilt $F(x) = \lim\limits_{n\to\infty} F(x_n)$ für jedes $x \in \mathbb{R}$ und jede Folge (x_n) mit $x_1 \leq x_2 \leq x_3 \leq ...$ und $\lim\limits_{n\to\infty} x_n = x$.

c) Es gelten: $\lim\limits_{n\to\infty} F(-n) = 0$, und $\lim\limits_{n\to\infty} F(n) = 1$.

2.3 Fehler bei Tests

Bei jedem Test gibt es zwei verschiedene Arten von Fehlern die auftreten können. Diese Fehler entstehen, da die wahre Verteilung P_ϑ sowohl von einem $\vartheta \in \Theta_0$, als auch von einem $\vartheta \in \Theta_1$ erzeugt werden könnte. Ist der wahre Parameter $\vartheta \in \Theta_0$ und es wird fälschlicherweise die Entscheidung für H_1 getroffen, spricht man von einem *Fehler erster Art*. Bei einem *Fehler zweiter Art* hingegen wird die Entscheidung für H_0 getroffen, obwohl der wahre Parameter $\vartheta \in \Theta_1$ ist.

Im Beispiel 2.3 würde es zu einem Fehler erster Art kommen, wenn wir zu dem Ergebnis kämen, dass der Würfel unfair ist, obwohl er eigentlich fair ist. Wir würden den Würfel also ungerechtfertigter Weise als unfair einstufen. Dieser Fehler erster Art ist meist schwerwiegender als ein Fehler zweiter Art. Bei einem Fehler zweiter Art würden wir zu dem Ergebnis kommen, dass der Würfel fair ist, obwohl dies in Wirklichkeit nicht der Fall ist, siehe Tabelle 1.

Testentscheidung	Realität: $\vartheta \in \Theta_0$	Realität: $\vartheta \in \Theta_1$
H_0	richtige Entscheidung	Fehler zweiter Art
H_1	Fehler erster Art	richtige Entscheidung

Tabelle 1: Resultate eines Tests

Da diese Fehler unvermeidbar sind, ist es ausschlaggebend für die Qualität des Testergebnisses mithilfe eines gut gewählten kritischen Bereichs die Wahrscheinlichkeiten für das Auftreten solcher Fehler möglichst gering zu halten. Umgekehrt ist es also nicht möglich eine Entscheidung für H_0 oder H_1 mit vollständiger Sicherheit zu treffen. Um stattdessen ein Maß für die Sicherheit der Entscheidung des Tests zu bestimmen, führen wir die *Gütefunktion* $g_K(\vartheta)$ ein.

Definition 2.3.1 (Gütefunktion)**.** Die *Gütefunktion* $g_K \colon \Theta \to [0,1]$ ordnet jedem $\vartheta \in \Theta$ die Wahrscheinlichkeit zu mit der die Nullhypothese H_0 verworfen wird, wenn die Verteilung P_ϑ vorliegt. Die Gütefunktion für einen Test mit kritischem Bereich K lautet:

$$g_K(\vartheta) \coloneqq P_\vartheta(S \in K), \text{ für alle } S \in \Omega^n.$$

Für einen perfekten Test mit einem perfekten kritischen Bereich K_{perfekt} würde für die Gütefunktion gelten:

$$g_{K_{\text{perfekt}}}(\vartheta) = \begin{cases} 0, & \text{falls } \vartheta \in \Theta_0 \\ 1, & \text{falls } \vartheta \in \Theta_1. \end{cases}$$

Diese Gütefunktion besitzt ein Test, der für jedes $\vartheta \in \Theta_0$ bzw. Θ_1 die richtige Entscheidung für H_0 bzw. H_1 wählt. Da dies jedoch in realen Testumgebungen unrealistisch ist, legt man eine obere Schranke $\alpha \in (0\%, 100\%)$ für die Wahrscheinlichkeit eines Fehlers erster Art fest, also für den Fall, dass die Nullhypothese $H_0 : \vartheta \in \Theta_0$ korrekt ist, aber vom Test abgelehnt wird. In einem Test mit einer solchen oberen Schranke α gilt also:

$$g_K(\vartheta) \leq \alpha, \text{ für jedes } \vartheta \in \Theta_0.$$

Solche Tests heißen *Signifikanzests zum Niveau* α oder kurz *Niveau* α*-Tests*. Das festgelegte Signifikanzniveau α führt bei mehrmaliger Testdurchführung dazu, dass maximal α der Tests zur Ablehnung von H_0 führen, wenn H_0 tatsächlich gilt. Ein Signifikanzniveau von $\alpha = 1\%$, würde dafür sorgen, dass bei korrekter Nullhypothese erwartungsgemäß einer aus 100 Tests H_0 fälschlicherweise ablehnt. Wählt man α also gering ($< 1\%$), erlaubt ein Testergebnis gegen H_0 begründete Zweifel an der Nullhypothese, da dieses Ergebnis nur in α der Tests auftreten würde, wenn H_0 tatsächlich gelten würde. Falls ein Niveau α-Test zu einer Ablehnung der Nullhypothese führt, sagt man: *Die Ablehnung von* H_0 *ist signifikant zum Niveau* α. Je kleiner das Niveau α gewählt wird, desto signifikanter ist im Falle einer Ablehnung von H_0 das Testergebnis.

Man erachtet also den Fehler erster Art als schwerwiegender und legt darum eine obere Schranke α für diesen fest. Dadurch ist man eher gewillt eine falsche Nullhypothese zu bestätigen (Fehler zweiter Art), als eine korrekte abzulehnen (Fehler erster Art). Entsprechend muss bei der Wahl der Hypothesen berücksichtigt werden, dass ein Fehler zweiter Art deutlich häufiger auftreten kann, als ein Fehler erster Art, da für diesen keine obere Schranke festgelegt ist. Deshalb eignen sich solche Tests dazu, die Alternativhypothese mit einer gewissen Signifikanz als wahr zu bezeichnen. Sie eignen sich jedoch nicht dazu, eine Nullhypothese als wahr zu identifizieren. Ein Testergebnis für H_0 hat lediglich die Aussage, dass H_0 nicht widerlegt werden konnte. H_0 wird dadurch in keinster Art und Weise bewiesen oder validiert.

Aus diesem Grund wählen wir in Beispiel 2.3 unsere Alternativhypothese H_1 als die Aussage, dass der Würfel häufiger als ein fairer Würfel die Sechs würfelt. Uns interessiert nämlich, ob man mit einer gewissen Signifikanz sagen kann, dass der Würfel unfair ist. Ein Nachweis der Nullhypothese, dass der Würfel nicht häufiger als ein fairer Würfel die Sechs würfelt, ist für uns vergleichsweise uninteressant, da dies kein außergewöhnlicher Umstand wäre.

Beispiel 2.4 (Rechtsseitiger Signifikanztest zum Niveau 10%)**.** Wir führen das Beispiel 2.3 fort und beschreiben unsere Teststatistik T, also die Anzahl der gewürfelten Sechsen, mit der Zufallsvariablen $Y \sim Bin\left(120, \frac{1}{6}\right)$. Anstatt jedoch einen kritischen Bereich zu bilden, indem man sich aus dem Sachkontext heraus einen sinnvollen kritischen Wert c überlegt, möchten wir nun einen Signifikanztest zum Niveau $\alpha = 10\%$ erstellen. Das heißt wir suchen den kritischen Wert c der in maximal 10% der Testfälle überschritten wird. Dadurch kann gewährleistet werden, dass es in maximal 10% der Tests zu einer Ablehnung der Nullhypothese kommt, wodurch wiederum in maximal 10% der Tests ein Fehler erster Art auftreten kann. Es soll also gelten: $P(Y \geq c) \leq 10\%$. In Worten: *Die Wahrscheinlichkeit häufiger als c-mal eine Sechs zu würfeln soll kleiner als* 10% *sein.* Da in

einer Binomialverteilung nur ganzzahlige Werte angenommen werden, erhalten wir folgende Werte für c, zwischen denen die 10% Grenze überschritten wird:

$$P(Y \geq 26) \approx 0.092$$
$$P(Y \geq 25) \approx 0.136$$

Wir wählen den kritischen Bereich also mit $K = \{Y \geq 26\}$, damit unser Signifikanzniveau $\alpha = 10\%$ nicht überschritten wird. Denn je größer α gewählt wird, desto weniger signifikant wäre ein Testergebnis gegen H_0. Daher nehmen wir den nächstkleineren Wert 9.2% und erhalten somit Abbildung 1 und einen kritschen Wert $c = 26$. Hier liegt der kritische Bereich ausschließlich auf der rechten Seite des Erwartungswertes, weshalb wir von einem *rechtsseitigen Test* sprechen. Sollte die Teststatistik in den orangen kritischen Bereich fallen, wird die Nullhypothese abgelehnt.

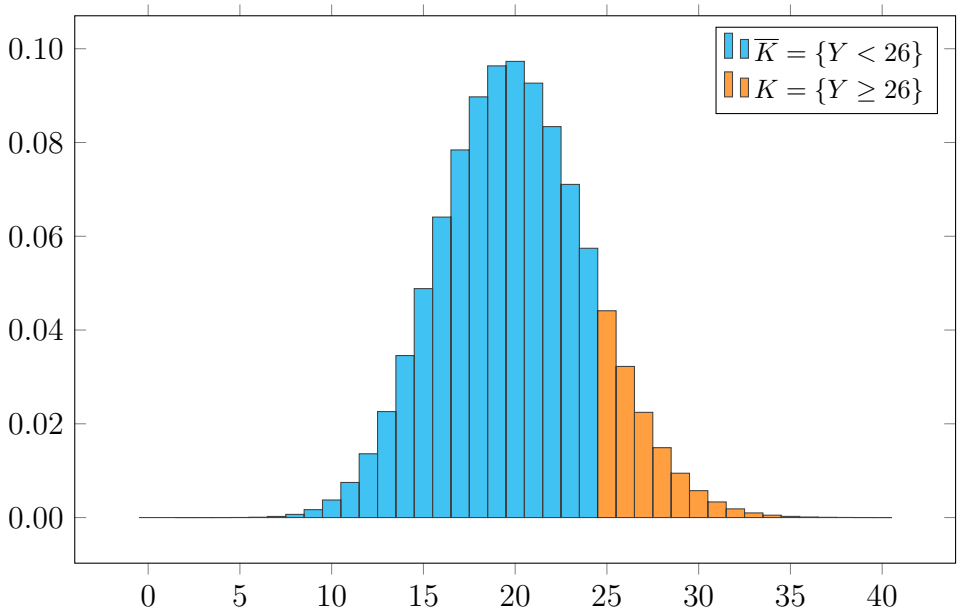


Abbildung 1: Wahrscheinlichkeitsverteilung von $Y \sim \text{Bin}(120, \frac{1}{6})$ mit kritischem Bereich K und Annahmebereich $\overline{K}$ für einen einseitigen Test zum Niveau 10%

Beispiel 2.5 (Zweiseitiger Signifikanztest zum Niveau 10%)**.** Wir haben in den bisherigen Beispielen nur Tests betrachtet, die prüfen sollen, ob der vorliegende Würfel häufiger als normal eine Sechs würfelt. Dementsprechend wurde die Alternativhypothese H_1 *einseitig nach oben* oder auch *rechtsseitig* formuliert, nämlich $H_1 : \vartheta > \frac{1}{6}$. Sollte uns nun interessieren ob der Würfel sich fair verhält, müssen wir einen sogenannten *zweiseitigen Test* durchführen. Wir wissen bereits, dass ein

fairer Würfel mit einer Wahrscheinlichkeit von $\frac{1}{6}$ eine Sechs würfelt. Jede andere Wahrscheinlichkeit betrachten wir als unfair. Für den zweiseitigen Test lautet dementsprechend die Nullhypothese $H_0 : \vartheta = \frac{1}{6}$. H_0 enthält also nur einen Wert für den unbekannten Parameter ϑ. Die Gegenhypothese lautet $H_1 : \vartheta \neq \frac{1}{6}$. Bei einem solchen zweiseitigen Test werden Werte der Teststatistik die nach oben oder nach unten stark vom Erwartungswert abweichen als Indiz für die Gegenhypothese betrachtet. Entsprechend legt man den kritischen Bereich, wie in Abbildung 2 zu sehen, auf beide Seiten des Erwartungswertes. Bei einem gleichbleibenden Testniveau $\alpha = 10\%$ teilen wir α auf beide Seiten des Histogramms auf, sodass links und rechts des Erwartungswertes jeweils ein 5% Bereich gewählt wird.

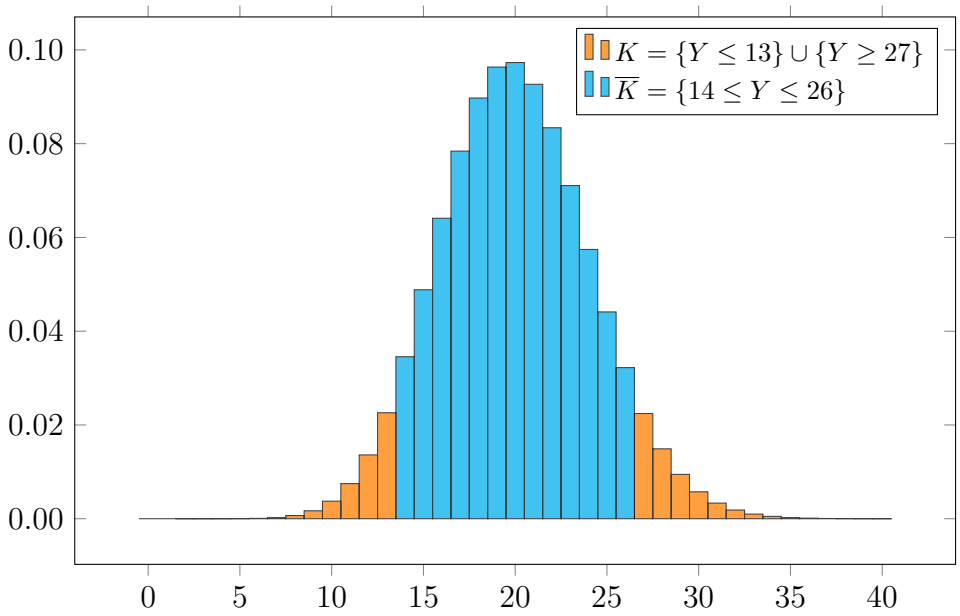


Abbildung 2: Wahrscheinlichkeitsverteilung von $Y \sim \text{Bin}(120, \frac{1}{6})$ mit kritischem Bereich K und Annahmebereich $\overline{K}$ für einen zweiseitigen Test zum Niveau 10%

2.4 Der p-Wert

Bisher haben wir zur Erstellung eines Signifikanztests immer ein Signifikanzniveau α gewählt. Dieses Niveau stellt eine obere Schranke für die Wahrscheinlichkeit eines Fehlers erster Art dar. Anhand dieser Schranke haben wir dann den kritischen Bereich gewählt, woraus sich mithilfe einer Stichprobe das Testergebnis und dessen Signifikanz ergab. Eine andere, häufiger verwendete Methode ist das Bestimmen des sogenannten *p-Wertes* $p(x)$ *zu einer Beobachtung* x, wobei x einen Wert der Teststatistik $T(S)$ beschreibt, welcher aus der Stichprobe S berechnet wird. Man

wählt sich nach wie vor ein festes Signifikanzniveau α, welches diesmal jedoch keinen Einfluss auf den kritischen Bereich hat, sondern später mit dem p-Wert verglichen wird. Wir definieren uns für die folgenden Aussagen ein klassisches Signifikanzniveau α^* (Definition 2.3.1), welches Einfluss auf den kritischen Bereich hat und nur für die Berechnung des p-Wertes relevant ist. „Der p-Wert $p(x)$ zur Beobachtung x ist die kleinste Zahl α^*, für die die Wahl von α^* als Testniveau ("gerade noch") zur Ablehnung von H_0 führt"[6, S. 281].

Nehmen wir einmal das Testproblem $L = (M, [0, \vartheta_0], (\vartheta_0, 1])$ mit $\vartheta_0 = \frac{1}{6}$ aus Beispiel 2.2 an. Sinnvolle kritische Bereiche für dieses Problem besitzen die Form $K = \{T(S) \geq c\}$. S seien n Ergebnisse eines Würfelwurfes und die Teststatistik $T(S)$ sei die Anzahl der in S enthaltenen Sechsen mit dem Wert $T(S) = l$. Um nun den p-Wert $p(l)$ zu ermitteln, betrachten wir alle kritischen Bereiche, die l enthalten und somit zu einer Ablehnung von H_0 führen. Der kleinste dieser Bereiche ist $K_l = \{l, l+1, ..., n\} = \{T \geq l\}$. Da dieser der kleinstmögliche sinnvolle Bereich der oben genannten Form $\{T(S) \geq c\}$ ist, besitzt auch das zugehörige Signifikanzniveau α^* den kleinstmöglichen Wert unter der Bedingung, dass H_0 mit $T(S) = l$ gerade noch abgelehnt werden soll.

Der p-Wert $p(l)$ entspricht also diesem kleinstmöglichen Signifikanzniveau α^*. Es gilt für einen Test mit $T(S) = l$:

$$p(l) = \alpha^* = P(S \in K) = P(T(S) \geq l)$$

Sollte also wie im Beispiel 2.4 ein Würfel darauf untersucht werden, ob er zu oft eine Sechs würfelt, dann würde man bei einem Stichprobenumfang von $n = 120$ und einer Anzahl von Sechsen von $T(S) = 27$ den p-Wert

$$p(27) = \sum_{j=l}^{n} \binom{n}{j} \vartheta_0^j (1-\vartheta_0)^{n-j} = \sum_{j=27}^{120} \binom{120}{j} \left(\frac{1}{6}\right)^j \left(\frac{5}{6}\right)^{120-27} \approx 0.0597$$

erhalten. Die Testentscheidung kann anschließend unmittelbar aus dem Vergleich von dem zu Anfang festgelegten Signifikanzniveau α und dem p-Wert getroffen werden:

$$f(S) = \begin{cases} 1 & \text{, falls } \alpha > p(x) \\ 0 & \text{, falls } \alpha \leq p(x). \end{cases}$$

Allgemein gilt auch hier: Je kleiner der p-Wert ist, desto signifikanter ist ein Testergebnis $f(S) = 1$, dass zur Ablehnung von H_0 führt. Man spricht auch hier von einem *Signifikanztest zum Niveau* α.

2.5 Verteilungen von Teststatistiken

Die Struktur eines Tests wird im Wesentlichen davon bestimmt, wie dessen Teststatistik definiert ist. Durch diese wird der kritische Bereich festgelegt, wodurch wiederum die Testfunktion definiert ist, welche schlussendlich das Testergebnis bestimmt. Teststatistiken selbst sind Zufallsvariablen, da sie aus einer Ergebnismenge (den möglichen Stichproben) auf einen reellen Wert abbilden. Somit können wir für jede Teststatistik auch einen Wahrscheinlichkeitsraum mit einer Wahrscheinlichkeitsverteilung definieren. Diese Verteilung der Teststatistik T nennt man *Referenzverteilung*.

Die Referenzverteilung kann etwa wie in Beispiel 2.3 die Binomialverteilung sein. Wenn dies der Fall ist, spricht man von einem *Binomialtest*. Eine Teststatistik kann auch andere Verteilungen besitzen. Bisher haben wir ausschließlich Verteilungen von diskreten Zufallsvariablen behandelt. Für viele Anwendungsbereiche reichen diese jedoch nicht aus. Darum führen wir nun *stetige Zufallsvariablen* ein.

Definition 2.5.1 (Stetige Zufallsvariable)**.** Eine Zufallsvariable X heißt stetig verteilt, wenn es eine nichtnegative integrierbare Funktion $f : \mathbb{R} \to \mathbb{R}$ mit der Eigenschaft

$$\int_{-\infty}^{\infty} f(t)\,dt = 1$$

gibt, sodass die Verteilungsfunktion F von X die Darstellung

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)\,dt, \text{ für } x \in \mathbb{R}$$

besitzt. In diesem Fall sagt man auch, *X hat eine stetige Verteilung.* Die Funktion f heißt *Dichte* von X bzw. *Dichte der Verteilungsfunktion von X*. Für die Verteilungsfunktion gilt für Werte $a, b \in \mathbb{R}$ mit $a < b$:

$$P(a \leq X \leq b) = F(b) - F(a) = \int_{a}^{b} f(t)\,dt.$$

Definition 2.5.2 (Normalverteilung)**.** Die stetige Zufallsvariable X hat eine *Normalverteilung* mit Erwartungswert μ und Standardabweichung σ^2 ($\mu \in \mathbb{R}, \sigma > 0$), falls X die Dichte

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \text{ mit } x \in \mathbb{R}$$

besitzt. Man schreibt kurz: $X \sim \mathcal{N}(\mu, \sigma^2)$. Die Dichte f besitzt eine symmetrische Glockenform, wobei die Stelle μ die Mitte dieser Symmetrie und gleichzeitig den Höhepunkt der Kurve kennzeichnet. Der Erwartungswert einer stetigen Zufallsvariable X berechnet sich mit

$$\mu = E(X) = \int_{-\infty}^{\infty} x \cdot f(x)\, dx, \text{ für } x \in X$$

und die Varianz mit

$$\sigma^2 = V(x) = \int_{-\infty}^{\infty} (x - E(X))^2 f(x)\, dx = E(X - E(X))^2.$$

Wie auch bei diskreten Zufallsvariablen nennt man die (positive) Wurzel $\sigma = \left|\sqrt{\sigma^2}\right|$ Standardabweichung von X.

Definition 2.5.3 (Halbnormalverteilung)**.** Sei X eine stetige Zufallsvariable mit $X \sim \mathcal{N}(0, \sigma)$. Die Zufallsvariable $Y = |X|$ heißt *halbnormalverteilt* mit Dichte

$$f(y) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \exp\left(-\frac{y^2}{2\sigma^2}\right), y > 0 \quad \text{(Abbildung 3)}$$

und Verteilungsfunktion

$$F(y) = \frac{2}{\sqrt{\pi}} \int_0^{\frac{y}{\sigma\sqrt{2}}} \exp(-t^2)\, dt.$$

Erwartungswert und Varianz sind definiert als

$$E(Y) = \frac{\sigma\sqrt{2}}{\sqrt{\pi}} \quad \text{und} \quad V(Y) = \sigma^2 \left(1 - \frac{2}{\pi}\right).$$

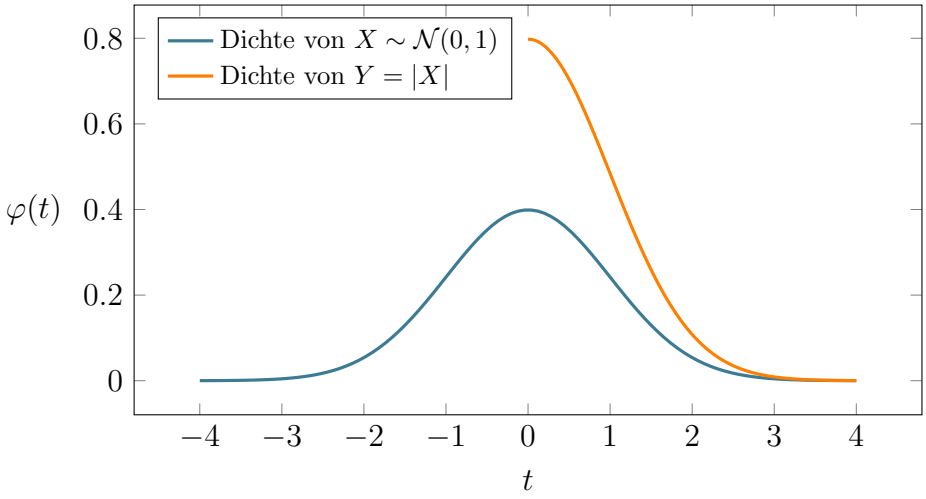
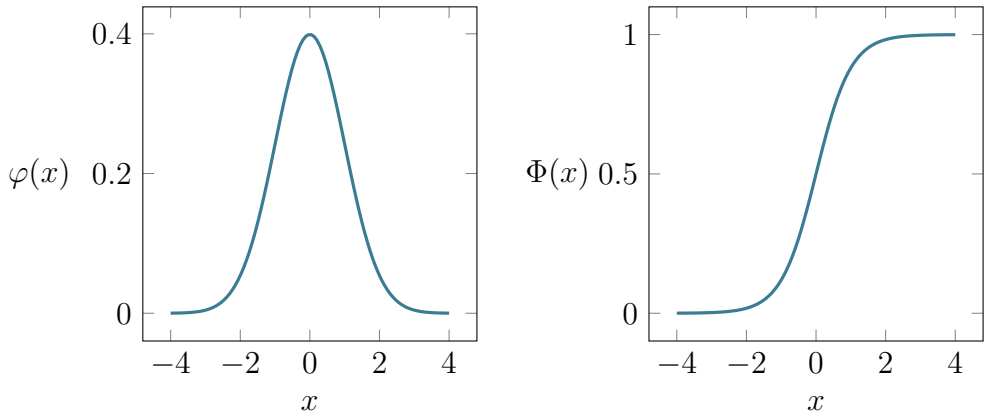


Abbildung 3: Dichtefunktion einer halbnormalverteilten Zufallsvariable Y und einer standardnormalverteilten Zufallsvariable X [13]

Eine besondere Rolle kommt der Normalverteilung mit den Werten $\mu = 0$ und $\sigma^2 = 1$ zu. Man spricht hier von der *standardisierten Normalverteilung* oder auch *Standardnormalverteilung*. Eine standardnormalverteilte Zufallsvariable besitzt die Dichtefunktion $\varphi(x)$ und Verteilungsfunktion $\Phi(x)$, wie zu sehen in Abbildung 4.



(a) $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ (b) $\Phi(x) = \int_{-\infty}^{x} \varphi(z)\, dz$

Abbildung 4: Dichte und Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen $X \sim \mathcal{N}(0,1)$

Man nutzt diese Standardnormalverteilung, um für alle beliebigen Normalverteilungen Werte der Verteilungsfunktion effizient und einheitlich berechnen zu

können. Das Problem an der Verteilungsfunktion $\Phi(x) = \int_{-\infty}^{x} \varphi(z)dz$ einer Normalverteilung ist das Bilden der Stammfunktion zum Lösen des Integrals über die Dichtefunktion $\varphi(x)$. Diese Stammfunktion existiert nicht in geschlossener Form, weshalb Werte von solchen Verteilungsfunktionen über *numerische Integration* berechnet werden müssen. Auf numerische Integration wird in dieser Arbeit nicht genauer eingegangen. Es reicht zu wissen, dass sie sehr aufwendig ist, weshalb man alle Normalverteilungen durch Standardisierung auf die Standardnormalverteilung zurückführt. Für diese existiert eine Standardtabelle (siehe Anhang A Abbildung 5), in welcher alle Werte von $\Phi(x)$ eingetragen sind. Um Werte einer Verteilungsfunktion einer Zufallsvariablen $X \sim \mathcal{N}(\mu, \sigma^2)$ mithilfe der Standardtabelle zu berechnen, standardisiert man X und erhält somit eine Zufallsvariable $Z \sim \mathcal{N}(0, 1)$ mit $Z = \frac{X-\mu}{\sigma}$. Hieraus ergibt sich

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right) = \Phi(z).$$

Der Wert von $\Phi(z)$ lässt sich nun aus der Standardtabelle ablesen. Werte mit $z < 0$ sind nicht eingetragen und lassen sich stattdessen aufgrund der Symmetrie der Verteilungsfunktion über $\Phi(-z) = 1 - \Phi(z)$ errechnen. Neben der naheliegenden Normalverteilung, lässt sich auch die Binomialverteilung annähernd durch die Standardnormalverteilung darstellen. Dies ist das Ergebnis des sogenannten *Zentralen Grenzwertsatz von de Moivre und Laplace.*

Satz 2.1 (Zentraler Grenzwertsatz (ZGWS) von de Moivre-Laplace)**.** Sei $S_n \sim \text{Bin}(n, p)$ eine Zufallsvariable mit $0 < p < 1$. Dann gilt für alle $a, b \in \mathbb{R}$ mit $a < b$:

$$\lim_{n\to\infty} P(a \leq \frac{S_n - np}{\sqrt{npq}} \leq b) = \int_a^b \varphi(x)\, dx, \tag{1}$$

$$\lim_{n\to\infty} P(\frac{S_n - np}{\sqrt{npq}} \leq b) = \int_{-\infty}^b \varphi(x)\, dx. \tag{2}$$

Für eine Zufallsvariable $X \sim \text{Bin}(n, p)$ lässt sich also für ausreichend große n ein Wert für $P(c \leq X \leq d)$ mit der Standardnormalverteilung annähern und aus der Standardtabelle (Tabelle 5 Anhang A) ablesen. Aus 1) folgt:

$$\begin{aligned} P(c \leq X \leq d) &= P\left(\frac{c-np}{\sqrt{npq}} \leq \frac{X-np}{\sqrt{npq}} \leq \frac{d-np}{\sqrt{npq}}\right) \\ &= \int_{\frac{c-\mu}{\sigma}}^{\frac{d-\mu}{\sigma}} \varphi(x)\, dx = \Phi\left(\frac{d-\mu}{\sigma}\right) - \Phi\left(\frac{c-\mu}{\sigma}\right) \end{aligned}$$

Eine allgemeinere Variante des ZGWS ist der *Zentrale Grenzwertsatz von Lindeberg-Lévy*. Hierbei muss keine binomialverteilte Zufallsvariable vorhanden sein, sondern lediglich eine Summe unabhängiger und identisch verteilter Zufallsvariablen.

Satz 2.2 (Zentraler Grenzwertsatz (ZGWS) von Lindeberg-Lévy)**.** Seien $X_1, ..., X_n$ stochastisch unabhängige und identisch verteilte Zufallsvariablen mit positiver Varianz $\sigma^2 = V(X_1)$. Setze man $\mu = E(X_1)$ und $S_n = X_1 + ... + X_n$, so gilt für all $a, b \in \mathbb{R}$ mit $a < b$:

$$\lim_{n\to\infty} P(a \leq \frac{S_n - n \cdot \mu}{\sigma\sqrt{n}} \leq b) = \int_a^b \varphi(x)\, dx, \tag{3}$$

$$\lim_{n\to\infty} P(\frac{S_n - n \cdot \mu}{\sigma\sqrt{n}} \leq b) = \int_{-\infty}^b \varphi(x)\, dx. \tag{4}$$

Die Aussage von Lindeberg-Lévy ist vollkommen unabhängig von der Verteilung von X_1 und lässt daher auf die Aussage von de Moivre-Laplace schließen, welche auf stochastisch unabhängige bernoulli-verteilte Zufallsvariablen spezifiziert ist. Um den ZGWS von Lindeberg-Lévy auf den von de Moivre-Laplace zurückzuführen, wählt man $X_1, ..., X_n$ mit $X \sim \text{Ber}(p)$ als stochastisch unabhängige und identisch verteilte Zufallsvariablen und erhält automatisch den ZGWS von de Moivre-Laplace mit $S_n \sim \text{Bin}(n, p)$. Grundsätzlich ist es für eine Zufallsvariable $X \sim \text{Bin}(n, p)$ für große n deutlich effizienter, Werte der diskreten Verteilungsfunktion der Form

$$F(x) = P(X \leq x) = ... + P(X = x - 2) + P(X = x - 1) + P(X = x)$$

mithilfe der Standardnormalverteilung anzunähern, anstatt jeden Summanden einzeln zu berechnen und anschließend die Summe zu bilden[2].

Anmerkung 2.1 (Weitere Referenzverteilungen)**.** *Es gibt noch viele weitere wichtige Referenzverteilungen für statistische Tests, wie etwa die sogenannte Chi-Quadrat-Verteilung oder die t-Verteilung. Sie werden in dieser Arbeit nicht thematisiert, spielen jedoch trotzdem eine wichtige Rolle in vielen statistischen Tests. Um ein Verständnis für statistisches Testen im Anwendungsbereich Kryptographie zu entwickeln, sind sie jedoch nicht zwingend notwendig. Daher gehen wir in den folgenden Kapiteln nicht näher auf den t-Test oder die Chi-Quadrat-Tests auf Anpassung, Unabhängigkeit und Homogenität ein[7, S. 247].*

2.6 Grundlagen der Kryptographie

Kryptographie kann als die Wissenschaft der Informationssicherheit verstanden werden. Sie liefert verschiedene Algorithmen und mathematische Verfahren um diese Sicherheit zu erreichen. Wichtige Aspekte sind *Daten-Vertraulichkeit, Daten-Integrität, Daten-Ursprungsauthentizität* und *Entitäts-Authentizität*[15, S. 4]. Diese beruhen auf den vier grundlegenden Zielen der Kryptographie. Aus diesen lassen sich alle weiteren Ziele ableiten und zusammen gewährleisten sie Informationssicherheit.

Definition 2.6.1 (Ziele der Kryptographie)**.** Die Ziele der Kryptographie sind:

1. **Daten-Vertraulichkeit:** Daten sollen ausschließlich von berechtigten Personen gelesen werden können.
2. **Daten-Integrität:** Daten sollen vor Veränderungen durch Dritte geschützt werden. Im Falle einer ungewollten Veränderung soll diese für berechtigte Personen erkennbar sein.
3. **Authentizität:** Beteiligte an einem Datenaustausch sollen sich gegenseitig eindeutig identifizieren können. Es soll klar erkennbar sein welchem Ursprung die Daten entstammen.
4. **Verbindlichkeit:** Jede Aktivität in Bezug auf ein Datum muss unanfechtbar einem Urheber zugeordnet werden können.

Ein wichtiges Konzept, das sich die Kryptographie zunutze macht, ist der Zufall. Im Verlauf dieser Arbeit wird die Definition und der Nutzen des Zufalls genauer erläutert. Zunächst führen wir einige Begriffe ein, die wir in diesem Kontext verwenden. Wir orientieren uns an den Definitionen aus dem *Handbook of Applied Cryptography* von Menezes, Oorschot und Vanstone[15].

Definition 2.6.2 (Bitgenerator)**.** Ein *Bitgenerator (Generator)* ist ein Verfahren, das eine Folge $b = b_1 b_2 ...$ aus Bits b_i mit $i \in 1, 2, ...$ ausgibt.

Definition 2.6.3 (Echter Zufallszahlengenerator)**.** Ein *echter Zufallszahlengenerator (TRNG)* ist ein Bitgenerator, dessen Verfahren ein physikalischer Prozess ist. Die Ausgabefolge b besteht aus stochastisch unabhängigen und gleichverteilten Bits b_i. Wir nennen die Ausgabe *echte Zufallszahl*.

Da solche TRNGs auf physikalischen Prozessen beruhen, wird ihre Ausgabefrequenz von der des physikalischen Prozesses begrenzt. Dies führt dazu, dass TRNGs für viele Anwendungsbereiche zu langsam oder auch *ineffizient* sind. Wir sprechen

von einem effizienten Prozess, Algorithmus oder Verfahren, wenn diese in Polynomialzeit arbeiten. Für das Generieren von vielen beziehungsweise langen Ausgaben nutzt man die effizienteren *Pseudozufallszahlengeneratoren*.

Definition 2.6.4 (Pseudozufallszahlengenerator)**.** Ein *Pseudozufallszahlengenerator (PRNG)* ist ein Bitgenerator, der aus einer echt zufälligen Eingabe der Länge l eine Ausgabe der Länge $k \gg l$ erzeugt. Die Ausgabe kann von keinem effizienten Algorithmus von einer echten Zufallszahl unterschieden werden. Wir nennen die Eingabe *Seed* und die Ausgabe *Pseudozufallszahl*.

PRNGs erhalten ihren Seed oft aus der Ausgabe eines TRNGs und erzeugen daraus ihrerseits effizient eine Ausgabe, die deutlich länger als der Seed ist. In gleicher Zeit generiert der PRNG also deutlich mehr Ausgaben als ein TRNG. Wir fassen beide Generatoren im Folgenden zu Zufallszahlengeneratoren (RNGs) zusammen. In dieser Arbeit betrachten wir hauptsächlich PRNGs, da sie aufgrund ihrer Effizienz weit verbreitet sind. Speziell *kryptographisch sichere Pseudozufallszahlengeneratoren* werden in der Praxis benötigt. Für deren Definition, definieren wir zunächst *vernachlässigbare* Funktionen.

Definition 2.6.5 (vernachlässigbar)**.** Eine Funktion $r : \mathbb{N} \to \mathbb{N}$ heißt *vernachlässigbar*, wenn für jedes Polynom $p : \mathbb{N} \to \mathbb{N}$ eine ganze Zahl k_0 existiert, sodass gilt:

$$r(k) < \frac{1}{p(k)}, \text{ für alle } k \geq k_0.$$

Definition 2.6.6 (Kryptographisch sicherer Pseudozufallszahlengenerator)**.** Ein *Kryptographisch sicherer Pseudozufallszahlengenerator (CSPRNG)* ist ein Pseudozufallszahlengenerator für dessen Ausgabe b gilt: Es gibt keinen effizienten Algorithmus, der mit der Eingabe der ersten $l-1$ Bits von b das l-te Bit von b mit einer Wahrscheinlichkeit größer $\frac{1}{2}+\varepsilon(x)$ bestimmen kann. Hierbei beschreibt ε eine vernachlässigbare Funktion.

Ein Beispiel für einen kryptographisch sicheren Pseudozufallszahlengenerator ist der *Micali-Schnoor Generator*. Er baut auf der sogenannten *RSA-Annahme* auf.

Definition 2.6.7 (RSA-Annahme)**.** Sei $n = p \cdot q$ eine positive ganze Zahl und das Produkt aus zwei verschiedenen Primzahlen p und q. Weiterhin sei e eine positive ganze Zahl mit $\mathrm{ggT}(e, (p-1)(q-1)) = 1$. Mit einer ganzen Zahl c wird angenommen: Es ist nicht effizient möglich eine ganze Zahl m zu finden, sodass gilt: $m^e \equiv c \mod n$.

Zusätzlich geht der Micali-Schnorr Generator noch davon aus, dass die Verteilung von $x^e \mod n$ für echt zufällige Bitsequenzen x der Länge r nicht effizient

von der Gleichverteilung auf dem Intervall $[0, n-1]$ unterschieden werden kann. Der vollständige Algorithmus des Generators wird in Algorithmus 1 beschrieben. Die genutzte Funktion ggT berechnet hierbei den größten gemeinsamen Teiler von zwei gegebenen Funktionsargumenten.

Algorithmus 1 : Micali-Schnorr pseudozufälliger Bitgenerator

Setup:

Wähle zwei verschiedene zufällige Primzahlen p und q.
Berechne $n = p \cdot q$ und $\phi = (p-1)(q-1)$.
Setze $N = \lfloor \lg n \rfloor + 1$.
Wähle e, mit $1 < e < \phi$, sodass $\text{ggT}(e, \phi) = 1$ und $80 \cdot e \leq N$.
Setze $k = \left\lfloor N(1 - \frac{2}{e}) \right\rfloor$ und $r = N - k$.

Generieren der pseudozufälligen Sequenz der Bitlänge $k \cdot r$:

Wähle eine echt zufällige Sequenz x_0 (den Seed) der Länge r.
Für i von 1 bis l:
    $y_i = x_{i-1}^e \mod n$.
    $x_i =$ die r höchstwertigen Bits von y_i.
    $z_i =$ die k niedrigstwertigen Bits von y_i.

Ausgabe: Sequenz $z_1, z_2, ..., z_l$.

Neben dem Seed x_0 der als echte Zufallszahl von einem TRNG geliefert wird, müssen viele der verwendeten Variablen in der Praxis noch weitere Eigenschaften erfüllen, damit der Generator kryptographisch sicher ist [15, S. 290]. Eine grundlegende Anforderung ist zum Beispiel, dass es nicht möglich sein darf n effizient in seine Primfaktoren p und q zu zerlegen. Damit dies gilt, müssen p und q entsprechend groß gewählt werden, um zu verhindern, dass p und q durch simples Ausprobieren erhalten werden können. Aus dem selben Grund sollte für p und q in etwa die gleiche Bitlänge gewählt werden. Gleichzeitig dürfen die Werte der beiden Primzahlen aber auch nicht zu ähnlich gewählt werden, da mit $p \approx q$ folgen würde: $n \approx p^2$. Dadurch könnte n vergleichsweise schnell durch Ausprobieren ermittelt werden und Rückschlüsse auf die Ausgaben könnten gezogen werden. Eine sinnvolle Parameterwahl ist daher für alle CSPRNGs essentiell um das nötige Maß an kryptographischer Sicherheit zu gewährleisten.

3 Statistische Tests und Kryptographie

Statistische Tests werden im Zusammenhang mit Kryptographie dazu benutzt, Zahlenfolgen zu untersuchen, die ein vermeintlicher Zufallszahlengenerator (RNG) generiert hat. Ziel ist es, die Zahlenfolgen als zufällig oder als nicht zufällig einstufen zu können. Je nachdem, welches Ergebnis der statistische Test liefert, kann mit einer gewissen Sicherheit gesagt werden, dass der zugrundeliegende Generator tatsächlich ein RNG ist, oder es eben nicht ist. Statistische Tests können jedoch nicht erkennen, ob ein getesteter Generator ein TRNG, PRNG oder CSPRNG ist. Wir gehen in dieser Arbeit davon aus, dass wir Algorithmen auf die Zufälligkeit ihrer Ausgabe testen, weshalb wir bei positivem Testergebnis von einem PRNG sprechen. Ein solcher erlaubt es uns zufällig wirkende Zahlenfolgen zu erzeugen, welche genutzt werden, um die Ziele der Kryptographie aus Definition 2.6.1 zu erreichen. Statistische Tests helfen uns, PRNGSs als solche zu identifizieren. Sie finden also überall dort Anwendung, wo PRNGs zur Generierung von Zufallszahlen gebraucht werden.

3.1 Zufallszahlen in der Kryptographie

Es gibt viele Algorithmen in der Kryptographie, die ohne zufällige Zahlenfolgen nicht funktionieren würden. Zufallszahlen bilden daher für viele Verfahren den Grundstein, weshalb es von enormer Wichtigkeit ist, dass diese Zahlenfolgen tatsächlich zufällig sind. Im Folgenden erläutern wir einige Anwendungsbeispiele von Zufallszahlen. In den jeweils angegebenen Quellen kann ein tieferes Verständnis für diese Konzepte entwickelt werden. Wir benutzen zur Veranschaulichung dieser Konzepte zwei Personen die miteinander sicher kommunizieren möchten. Wir nennen sie Alice und Bob und kürzen ihre Namen mit A und B ab.

3.1.1 Verschlüsselung

Für das erste Ziel der Kryptographie, Vertraulichkeit, werden Daten meist *verschlüsselt*. Dadurch können diese Daten im Idealfall nur von Personen *entschlüsselt* und somit gelesen werden, die den sogenannten *Schlüssel* besitzen. Beim Senden von Nachrichten soll so dafür gesorgt werden, dass nur Sender und Empfänger oder andere Berechtigte den Inhalt der Nachricht lesen können. Wir gehen im Folgenden genauer auf das „One Time Pad“ als Verschlüsselungsverfahren ein.

Definition 3.1.1 (Ver- und Entschlüsselung mit dem One Time Pad)**.** Seien $M, C, K = \{0,1\}^i$, mit $i \in \mathbb{R}$. Man nennt M die Klartextmenge, C die Geheimtextmenge und K die Schlüsselmenge[15, S. 21]. Elemente der jeweiligen Menge heißen

entsprechend Klartext m, Geheimtext c oder Schlüssel k. Eine *Verschlüsselung mit dem Schlüssel e* ist eine Funktion $E_k : M \to C$ mit $k \in K$. Für das One Time Pad gilt:

$$E_k(m) = m \oplus k = c,$$

wobei der Operator $\oplus$ eine bitweise Addition mod 2 der beiden Operanden durchführt. Eine *Entschlüsselung mit dem Schlüssel k* ist analog dazu eine Funktion $D_k : C \to M$. Für das One Time Pad gilt:

$$D_k = c \oplus k = m.$$

In Falle des One Time Pads wird k sowohl zum Ver- als auch zum Entschlüsseln genutzt, weshalb man von einer *symmetrischen Verschlüsselung* spricht. Werden zum Ver- und Entschlüsseln verschiedene Schlüssel verwendet, spricht man von einer *asymmetrischen Verschlüsselung*. Um nun einen vertraulichen Nachrichtenaustausch zwischen zwei Personen, Alice und Bob, auszuführen, einigen sich beide auf einen Schlüssel k. Alice möchte nun eine Nachricht $m \in M$ an Bob senden. Dafür verschlüsselt sie ihre Nachricht mit c = $E_k(m)$ und sendet sie Bob. Wenn diese Geheimtextnachricht c nun von einem unbefugten Dritten abgefangen wird, kann dieser ohne den Schlüssel k die Klartextnachricht m nicht ohne Weiteres lesen. Bob empfängt die Geheimtextnachricht c und entschlüsselt sie mit $D_k(c) = m$.

Obwohl das Verfahren des One Time Pads öffentlich bekannt ist, findet es dennoch Nutzen in der Kryptographie. Dieser stützt sich auf das *Kerckhoffs'sche Prinzip*. Es besagt, dass die Sicherheit eines Verschlüsselungsverfahrens auf der Geheimhaltung des Schlüssels basiert und nicht auf der Geheimhaltung des zugrundeliegenden Algorithmus[9, S. 6]. Um das Verfahren des One Time Pads sicher zu halten, wird für jede Nachricht ein neuer Schlüssel verwendet. Somit kann ein Angreifer, der einen Schlüssel einer älteren Nachricht besitzt, diesen nicht direkt nutzen, um zukünftige Nachrichten zwischen Alice und Bob zu entschlüsseln. Wichtig ist hierbei, dass die verschiedenen Schlüssel keine Abhängigkeiten erkennen lassen. Sonst wäre es einem Angreifer beispielsweise möglich aus alten Schlüsseln auf einen aktuellen zu schließen. Um die Anforderungen des One Time Pads an einen Schlüssel zu erfüllen, kann man diese von Zufallszahlengeneratoren erzeugen lassen. Bei vielen Nachrichten werden auch viele Schlüssel benötigt, weshalb man hier oft Pseudozufallszahlengeneratoren verwendet. Diese sind im Vergleich zu echten Zufallszahlengeneratoren meist effizienter[15, S. 11].

3.1.2 Digitale Signaturen

Digitale Signaturen werden dazu genutzt, Daten, wie etwa Nachrichten, zu *signieren* und zu *verifizieren*. Durch diese Verfahren kann die Urheberschaft einer Nachricht eindeutig demjenigen zugeordnet werden, der die Nachricht signiert hat. Wir erreichen damit die Ziele Authentizität und Verbindlichkeit unserer Definition 2.6.1. Zum Signieren einer Nachricht m benötigt man einen privaten Schlüssel k, welcher nur der signierenden Person Alice bekannt ist. Eine Signatur ist sowohl von dem Schlüssel, als auch von der Nachricht abhängig, die gesendet werden soll. Vor dem eigentlichen Signieren wird die Nachricht m mithilfe einer sogenannten *Hashfunktion* h auf einen Hashwert $h(m) = \widetilde{m}$ reduziert. Eine Hashfunktion besitzt unter anderem die Eigenschaft der Kollisionsresistenz. Das heißt, es ist nicht möglich zwei Eingabewerte mit dem gleichen Ausgabewert zu finden. Mithilfe des Hashwertes und des öffentlichen Schlüssels wird nun über eine Funktion $S_{A,k}$ die Signatur s berechnet: $S_{A,k}(\widetilde{m}) = s$. Üblicherweise ist $S_{A,k}$ das Inverse einer Falltürfunktion. Das heißt also die Berechnung ist ohne den verborgenen Parameter k nicht effizient möglich. Dies garantiert, dass nur A ihre Signatur herstellen kann, da sie als einzige ihren privaten Schlüssel besitzt. Alice sendet ihre Nachricht m mit der Signatur s an Bob. Dieser kann die Signatur nun verifizieren, indem er die öffentliche Verifizierungsfunktion V_A von Alice auf den Hashwert der Nachricht und die Signatur anwendet. Durch die Kollisionsresistenz der Hashfunktion ist es nicht effizient möglich eine zweite Nachricht passend zu der bereits erstellten Signatur zu verfassen. Die Verifizierungsfunktion V_A ist die Falltürfunktion zu der $S_{A,k}$ die Inverse ist. Das heißt das Verifizieren mit V_A ist ohne Probleme möglich. V_A gibt als Ergebnis entweder wahr oder falsch zurück, je nachdem ob die Signatur zur Nachricht und zu Alice passt oder nicht[15, S. 429].

Algorithmus 2 : Signieren und Verifizieren

Funktion : A signiert eine Nachricht m mit einer Signatur s. Person B verifiziert anschließend die Signatur.

1. Signieren
   (a) Wähle den privaten Schlüssel k.
   (b) Berechne $\widetilde{m} = h(m)$ und $s = S_{A,k}(\widetilde{m})$.
   (c) A's Signatur für m ist s.
2. Verifizieren
   (a) Berechne $\widetilde{m} = h(m)$ und $u = V_A(\widetilde{m}, s)$.
   (b) Akzeptiere Signatur s, wenn $u = wahr$.

Um dieses System zu missbrauchen, könnte ein Angreifer den privaten Schlüssel von Alice herausfinden und sich dann als Alice ausgeben[3]. Um diesen Missbrauch

schwieriger zu gestalten kann man daher auch hier die privaten Schlüssel mithilfe von Zufallszahlengeneratoren erstellen. Einige Signaturverfahren, wie DSA (Digital Signature Algorithm), generieren zusätzlich für jede Nachricht einmalige zufällige Zahlen beim Signieren[10, S. 15]. Auch hier wird also pro Nachricht eine Zufallszahl benötigt, weshalb man oft auf PRNGs zur effizienten Generierung zurückgreift.

3.2 Anforderungen an Zufallszahlen

Der Begriff des Zufalls ist laut dem deutschen Duden „etwas, dass man nicht vorausgesehen hat, was nicht beabsichtigt war, was unerwartet geschah“[8]. In Bezug auf eine Zahl bedeutet dies, dass man nicht vorhersehen kann, welchen Wert sie besitzt.

Betrachten wir einmal die Binärzahl b. Da wir nicht wissen welchen Wert b besitzt, können wir auch über kein Bit von b eine Aussage treffen. Vergleichbar ist dies mit dem mehrmaligen Werfen einer fairen Münze, wobei eine Seite der Münze „0“ und die andere Seite „1“ zeigt. Die Wahrscheinlichkeit für die beiden Ergebnisse ist jeweils $\frac{1}{2}$. Jedes Bit von b stellt einen Münzwurf dar und hängt ausschließlich von dem Ergebnis dieses aktuellen Münzwurfs ab. Mit einem solchen Verfahren kann man unter der Voraussetzung einer fairen Münze echte Zufallszahlen erzeugen. Solche echt zufälligen Folgen enthalten im Durchschnitt immer die gleiche Anzahl von 0- und 1-Bits. Die Eigenschaften des mehrmaligen unabhängigen Münzwurfs nutzen wir als Maßstab für die Betrachtung von Zufallszahlen.

Ausgehend von diesem Maßstab sollten PRNGs ihre aktuelle Ausgabe vollkommen unabhängig von vorausgegangenen Ausgaben wirken lassen. Zusätzlich darf keine Ausgabe des Generators auf den Seed schließen lassen. Dies ist besonders wichtig, da die PRNG-Algorithmen selbst meist öffentlich bekannt sind, wodurch die Vorhersehbarkeit der Ausgabebits nur durch die Geheimhaltung des Seeds verhindert wird. Daher wird der Seed echt zufällig generiert[1, S. 1–1].

3.3 Testen von Zufallszahlen

Ausgehend von den Anforderungen an Zufallszahlen aus Kapitel 3.2, definieren wir im Folgenden die Ausgangslage für das Testen von Zufallszahlen. Jedes Bit b_i, das von einem RNG als Teil einer Folge b generiert wird, ist gleichzusetzen mit einer Zufallsvariablen

$$b_i \sim \text{Ber}\left(\frac{1}{2}\right).$$

Diese b_i sind also identisch verteilt. Aus diesem Grund ist die Zahlenfolge $b = b_1 b_2 ... b_n \in \{bin(0), bin(1), ..., bin(2^n)\}$ mit Länge n gleichverteilt mit

$$P(b = z) = \frac{1}{|\{bin(0), bin(1), ..., bin(2^n)\}|} = \frac{1}{2^n}. \tag{5}$$

Die Bezeichnung $bin(k)$ beschreibt den Binärwert der Zahl k und ist nicht zu verwechseln mit der Schreibweise der Binomialverteilung. Da alle Bits zusätzlich unabhängig voneinander wirken sollen, sind alle b_i außerdem stochastisch unabhängig voneinander verteilt. Die Anzahl X der Bits mit Wert $b_i = 1$ kann daher als binomialverteilte Zufallsvariable

$$X \sim \text{Bin}\left(n, \frac{1}{2}\right)$$

angenommen werden. Wie am Ende von Kapitel 2.5 beschrieben, ist es für eine binomialverteilte Zufallsvariable mit großem n jedoch ineffizient die Verteilungsfunktion zu berechnen. Aus diesem Grund werden binomialverteilte Teststatistiken in Tests von Zufallszahlen meist zu standardnormalverteilten Zufallsvariablen umgeformt. Dies gelingt aufgrund der Aussagen der Zentralen Grenzwertsätze (2.1 und 2.2).

Für jeden statistischen Test, der eine Zahlenfolge b auf Zufälligkeit prüfen soll, beschreibt die Nullhypothese H_0 immer: $H_0 : \vartheta = \frac{1}{2^n}$, wobei b gleichverteilt ist mit $P(b = z) = \vartheta$. Einem Test geht immer eine Annahme voraus. Diese Annahme lautet: Eine Bitfolge b für die H_0 gilt, besitzt eine Eigenschaft E. Es wird also von der Zufälligkeit einer Bitsequenz auf eine andere Eigenschaft E geschlossen. Um H_0 zu testen, wird im Test selbst die Eigenschaft E getestet, um dann Rückschlüsse für H_0 zu ziehen. Die Gegenhypothese H_1 unterstellt der Folge entsprechend eine Nicht-Zufälligkeit und die Eigenschaft $\overline{E}$ mit $H_1 : \vartheta \neq \frac{1}{2^n}$. Das Testproblem für solche Tests lautet $\left(M, \left\{\frac{1}{2^n}\right\}, \left\{\vartheta | \vartheta \neq \frac{1}{2^n}\right\}\right)$, mit dem statistischen Modell $M = \left(\{0,1\}^n, P\right)$, wobei P die Gleichverteilung der Binärzahlen der Länge n beschreibt, siehe (5).

4 NIST Statistical Test Suite

Das „National Institute of Standards and Technology“(kurz: NIST) ist eine Bundesbehörde der Vereinigten Staaten von Amerika, die sich unter anderem mit der Standardisierung von Prozessen der Informationssicherheit und -verarbeitung befasst. Es veröffentlichte die NIST Statistical Test Suite erstmals am 15. Mai 2001 in dem Paper „A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications“ und aktualisierte sie zuletzt am 16. September 2010[1]. Die 15 enthaltenen statistischen Tests sollen als Grundlage für die Beurteilung der Eignung von Generatoren für die Kryptographie dienen. Hierbei schreibt das NIST seinen Tests keine Universalität zu. Es wird darauf hingewiesen, dass kein Set von Tests ausreicht, um die Eignung eines Generators in einer bestimmten Anwendungsumgebung zweifellos zu verifizieren.

Im Folgenden werden wir drei Tests im Detail betrachten und an Beispielen veranschaulichen. Hierbei beschreiben wir die Tests jeweils für die Eingabe von einer Bitfolge $b = b_1b_2b_3...b_n$ der Länge n. Das Ergebnis des Tests einer einzelnen Bitfolge fällt immer nach folgender Testfunktion aus:

$$f(b) = \begin{cases} 1 & \text{, falls } \alpha > p(T(b)) \\ 0 & \text{, falls } \alpha \leq p(T(b)). \end{cases}$$

Standardmäßig ist $\alpha = 0.01$ gesetzt. Der Hauptteil eines Tests befasst sich mit der Berechnung der Teststatistik $T(b)$ und dessen Verteilung. Daher liefern wir in den folgenden drei Betrachtungen keine explizite Testfunktion der Form von $f(b)$, sondern eine Beschreibung der Funktionsweise des jeweiligen Tests.

4.1 Frequency (Monobits) Test

4.1.1 Zweck des Tests

Der „Frequency (Monobits) Test“ ist der wohl grundlegendste Test auf Zufälligkeit. Die Annahme des Tests ist, dass zufällige Zahlen in etwa die gleiche Anzahl an 0- und 1-Bits besitzen. Die zu testende Eigenschaft E ist:„Etwa die Hälfte der Bits von b ist vom Wert 1.“ Entsprechend enthält eine zufällige Bitsequenz der Länge n etwa $\frac{n}{2}$ 1-Bits. In der „NIST Statistical Test Suite“ hängen alle anderen Tests von dem Ergebnis dieses Tests ab. Eine Sequenz b, die diesen Test nicht besteht, also das Testergebnis $f(b) = 1$ liefert, wird mit hoher Wahrscheinlichkeit auch keinen der folgenden Tests auf Zufälligkeit bestehen.

4.1.2 Testbeschreibung

Ausgehend von stochastisch unabhängigen bernoulli verteilten Bits b_i der Sequenz b, erzeugt man eine normalverteilte Teststatistik, um den p-Wert $p(b)$ effizient annähern zu können. Hierzu definiert man zunächst Zufallsvariablen $X_i = 2b_i - 1$. Jedes Bit $b_i = 0$ wird also zu $X_i = -1$ und jedes Bit $b_i = 1$ zu $X_i = 1$ umgewandelt. Aus den bernoulli verteilten $b_i \sim \text{Ber}\left(\frac{1}{2}\right)$ ergeben sich Zufallsvariablen $X_i : \{0,1\} \to \{-1,1\}$ mit einer Verteilung:

$$P(X_i = k) = \begin{cases} -1, & \text{für } k = 0 \\ 1, & \text{für } k = 1. \end{cases}$$

Da es sich bei X_i um identisch verteilte, diskrete Zufallsvariablen handelt, ergibt sich der Erwartungswert

$$E(X_i) = E(X_1) = \frac{1}{2}P(X = 1) + \frac{1}{2}P(X = -1) = 0.$$

Folglich erhält man für die Varianz

$$V(X_i) = V(X_1) = E((X_1 - E(X_1))^2) = E(X_1^2) = 1.$$

Mit dem Zentralen Grenzwertsatz von Lindeberg-Lévy 2.2 ergibt sich aus den stochastisch unabhängigen und identisch verteilten Zufallsvariablen $X_1, ..., X_n$ und der Summe $S_n = X_1 + ... + X_n$ eine Standardnormalverteilung für

$$\frac{S_n - n \cdot E(X_n)}{\sqrt{V(X_n)}\sqrt{n}} = \frac{S_n - n \cdot 0}{\sqrt{n}} = \frac{S_n}{\sqrt{n}}.$$

Man wählt $T(b) = \frac{S_n}{\sqrt{n}}$. Es gilt also $T(b) \sim \mathcal{N}(0,1)$. Für eine Beispielsequenz $b = 101010100$ ergibt sich mit $n = 10$:

$$\begin{aligned} T(b) &= \frac{1 + (-1) + (-1) + 1 + (-1) + 1 + (-1) + 1 + (-1) + (-1)}{10} \\ &= \frac{-2}{\sqrt{10}} = -0.632455532. \end{aligned}$$

Nun könnte man aus dieser Teststatistik den p-Wert

$$p\left(\frac{-2}{\sqrt{10}}\right) = P\left(\frac{-2}{\sqrt{10}} \leq T(b) \leq \frac{2}{\sqrt{10}}\right)$$

berechnen, indem man einen zweiseitigen Test für die Normalverteilung von T(b) ausführt. In der „NIST Statistical Test Suite“ wird allerdings der Betrag der Teststatistik betrachtet. $|T|$ ist mit $T \sim \mathcal{N}(0,1)$ eine halbnormalverteilte Zufallsvariable. Hierdurch wird deutlich, dass ausschließlich die Abweichung von T vom

Erwartungswert 0 von Interesse ist. Man erhält also einen einseitigen Test für die halbnormalverteilte Teststatistik $|T(b)|$ mit Verteilungsfunktion F, dessen p-Wert für die Beispielsequenz berechnet werden kann:

$$\begin{aligned} p(|T(b)|) &= p(|-0.632455532|) = P(|T| \geq 0.632455532) \\ &= 1 - P(|T| \leq 0.632455532) \\ &= 1 - F(0.632455532) \\ &= 1 - \frac{2}{\sqrt{\pi}} \int_0^{\frac{0.632455532}{\sigma\sqrt{2}}} \exp(-t^2)\, dt \\ &= 0.527089. \end{aligned}$$

Der Vergleich von p-Wert und Signifikanzniveau α ergibt anschließend:

$$\alpha = 0.01 \leq 0.527089 = p(|T(b)|).$$

Das Testergebnis ist daher eine Entscheidung für H_0. Die Aussage des Tests ist, dass die betrachtete Bitsequenz b keinen Widerspruch zur anfangs formulierten Eigenschaft E geliefert hat. Daraus folgt also auch kein Widerspruch zur Zufälligkeit der Bitsequenz und des zugrundeliegenden Zufallszahlengenerators.

4.2 Maurer's „Universal statistical" Test

4.2.1 Zweck des Tests

Der Maurer's "Universal statistical" Test wurde 1922 von Ueli M. Maurer entwickelt[14]. Als grundsätzliche Annahme geht Maurer davon aus, dass Sequenzen, die nicht komprimierbar sind, auch zufällig sind. Die zu testende Eigenschaft E ist: „Die Sequenz b ist nicht komprimierbar." Wir definieren nicht komprimierbare Sequenzen mithilfe der *Kolmogorov Komplexität*[11].

Definition 4.2.1 (Kolmogorov Komplexität)**.** Sei b eine Zeichenkette. Wir nennen $K(b)$ die Kolmogorov Komplexität von b mit:

$$K(b) = \text{„das kürzeste Programm in binär, das } b \text{ generiert."}$$

Definition 4.2.2 (Komprimierte Bitsequenzen)**.** Eine Bitsequenz b ist komprimiert oder auch nicht komprimierbar, genau dann, wenn gilt: $|K(b)| \geq |b|$.

Maurer nutzt als Maßstab für die Länge einer komprimierten Bitsequenz die Anzahl von Bits zwischen übereinstimmenden Mustern in der Sequenz. Als Muster verstehen wir eine festgelegte Folge von Bits von fester Länge. Je größer die Anzahl Bits zwischen übereinstimmenden Mustern ist, desto länger ist die entsprechende komprimierte Sequenz. Und desto wahrscheinlicher ist es, dass die Sequenz tatsächlich zufällig ist.

4.2.2 Testbeschreibung

Die Teststatistik $T(b)$ ist die Summe der $\log_2$-Distanzen zwischen allen jeweils übereinstimmenden Mustern, also die Anzahl Bits zwischen den Mustern. Wir definieren $T(b)$ später ausführlicher. Als Referenzverteilung wird auch hier die Halbnormalverteilung verwendet. Der Test benötigt eine Sequenzlänge der Größenordnung $10 \cdot 2^L + 1000 \cdot 2^L$ mit $6 \leq L \leq 16$. Diese recht spezifische Anforderung kommt daher, dass die Sequenz in zwei Segmente unterteilt wird: Das Initialisierungssegment, bestehend aus Q L-Bit Blöcken, und das Testsegment, bestehend aus K L-Bit Blöcken. Hierbei gilt dann etwa $Q = 10 \cdot 2^L$ und $K \approx 1000 \cdot 2^L$. In jedem Fall wird $K = \left\lfloor \frac{n}{L} \right\rfloor - Q$ gewählt, um die Zahl an L-Blöcken zu maximieren. Alle übrigen Bits am Ende der Sequenz werden nicht benutzt.

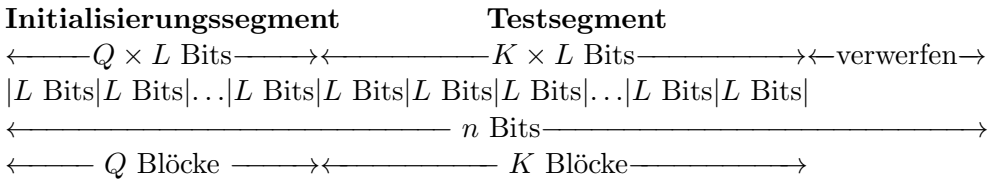


Abbildung 5: Aufteilung der Bitsequenz

Wir übernehmen das Beispiel der „NIST Test Suite“ und simulieren den Test an der Sequenz $b = 01011010011101010111$ mit den Werten $L = 2$ und $Q = 4$. Folglich gilt $K = \left\lfloor \frac{n}{L} \right\rfloor - Q = \left\lfloor \frac{20}{2} \right\rfloor - 4 = 6$. Diese Werte sind für reale Anwendungen nicht geeignet und dienen nur der Veranschaulichung. Das Initialisierungssegment ist also 01011010 und das Testsegment 011101010111. Die L-Bit Blöcke ergeben sich wie folgt:

Block	Segment	Muster
1	Initialisierungs-segment	01
2		01
3		10
4		10
5	Test-segment	01
6		11
7		01
8		01
9		01
10		11

Tabelle 2: L-Bit Blockaufteilung von b

Mithilfe des Initialisierungssegments wird nun die folgende Tabelle 3 erstellt und die erste Reihe wird initialisiert. Hierfür bekommt jedes mögliche L-Bit Muster (00, 01, 10 und 11) eine Spalte mit der Bezeichnung T_j, wobei j den Dezimalwert des L-Bit Musters angibt. Initialisiert wird jede Spalte mit dem Block i, in dem das jeweilige Muster zuletzt aufgetreten ist. Der Wert von Q sollte so groß gewählt werden, dass alle möglichen Muster auch im Initialisierungssegment auftreten. Dies ist im Beispiel nicht der Fall, wodurch am Anfang für T_0 und T_3 nicht der Abstand von Muster zu Muster betrachtet wird, sondern der Abstand vom Anfang der Sequenz zum ersten Auftreten des Musters. Dies verfälscht die Teststatistik T. Hier werden diese Spalten einfach mit 0 initialisiert. Anschließend wird das Testsegment Block für Block durchlaufen. Die Variablen T_i, mit tiefgestellten Indizes, bezeichnen Tabellenwerte, während die Variable T, ohne tiefgestellten Index, nach wie vor die Teststatistik bezeichnet.

Blocknummer	**Mögliche L-bit Werte für Muster**			
	T_0 **für 00**	T_1 **für 01**	T_2 **für 10**	T_3 **für 11**
Initialisierung	0	2	4	0
5	0	5	4	0
6	0	5	4	6
7	0	7	4	6
8	0	8	4	6
9	0	9	4	6
10	0	9	4	10

Tabelle 3: T_j-Werte nach Durchlauf aller Blöcke

Der erste Block des Testsegments im Beispiel ist Block 5. Er enthält das Mus-

ter 01. Daher wird die Spalte T_1 aktualisiert und die 2 wird durch eine 5 ersetzt. Diese Distanz zwischen dem Auftreten des Musters in Block 2 und Block 5 wird in einer Variablen S gespeichert: $S = \log_2(5-2) = 1.584962501$. In jedem folgenden Schritt wird für das jeweils auftretende Muster genau diese Distanz berechnet und auf die bisherige Summe S aufaddiert. Für die Blöcke $6-10$ folgt:

Block 6 enthält das Muster 11. Setze $T_3 = 6$ und berechne
$S = 1.584962501 + \log_2(6-0) = 4.169925002$.

Block 7 enthält das Muster 01. Setze $T_1 = 7$ und berechne
$S = 4.169925002 + \log_2(7-5) = 5.169925002$.

Block 8 enthält das Muster 01. Setze $T_1 = 8$ und berechne
$S = 5.169925002 + \log_2(8-7) = 5.169925002$.

Block 9 enthält das Muster 01. Setze $T_1 = 9$ und berechne
$S = 5.169925002 + \log_2(9-8) = 5.169925002$.

Block 10 enthält das Muster 11. Setze $T_3 = 10$ und berechne
$S = 5.169925002 + \log_2(10-6) = 7.169925002$.

Nach Durchlauf der Blöcke des Testsegments wird die Summe S durch die Anzahl der Blöcke K geteilt, die betrachtet wurden. Die Teststatistik ergibt sich aus:

$$T(x) = \frac{S}{K} = \frac{1}{K} \sum_{i=Q+1}^{Q+K} \log_2(i - T_j).$$

T_j bezeichnet hierbei jeweils den Wert T_j, der durch Block i in Zeile i überschrieben wird. Für das Beispiel ergibt sich ein Wert von

$$T(01011010011101010111) = \frac{7.169925002}{6} = 1.1949875.$$

Für den allgemeinen Fall der Berechnung der Teststatistik $T(b)$ wird folgender Algorithmus ausgeführt. Die Variablen T_{m_i} beschreiben den Wert in Zeile i und Spalte m_i, wobei m_i das Muster des Blockes i darstellt.

Algorithmus 3 : Berechnen der Teststatistik $T(b)$ für „Maurers Universal Statistical Test“

Eingabe : Eine binäre Bitsequenz $b = b_0 b_1 ... b_{n-1}$ der Länge n, und Parameter L, Q, K.

Ausgabe : Der Wert der Teststatistik $T(b)$ für die Sequenz b.

1: Für j von 0 bis $2^L - 1$: Setze $T_j = 0$.
2: Für i von 1 bis Q: Setze $T_{m_i} = i$.
3: Setze die Summe $S = 0$.
4: Für i von $Q + 1$ bis $Q + K$:
5: $\quad S = S + \log_2(i - T_{m_i})$.
6: $\quad T_{m_i} = i$.
7: Setze $T = S/K$.

Aus dieser Teststatistik berechnet man nun den p-Wert, um somit ein Testergebnis zu erhalten. Als Referenzverteilung verwendet der Test die Halbnormalverteilung mit Verteilungsfunktion F (Definition 2.5.3). Diese lässt sich jedoch nur für Zufallsvariablen der Form $Y = |X|$ mit $X \sim \mathcal{N}(0, \sigma^2)$ anwenden. Deshalb standardisiert man T und erhält somit für T^*:

$$T^* = \frac{T - \mu}{\sigma} = \frac{T - 1.5374383}{\sqrt{1.338}}.$$

Die Werte für μ und σ stammen aus der von Menezes, Oorschot und Vanstone erstellten Tabelle[15]. Diese Werte werden in Abhängigkeit von der Länge L des Musters gewählt. Maurer selbst lieferte für seinen Test zwar auch entsprechende Werte, allerdings nur für den von ihm als sinnvoll erachteten Bereich von $6-16$[14]. $T^*(b)$ ergibt sich durch die Standardisierung von $T(b)$:

$$T(b)^* = \frac{T(b) - \mu}{\sigma} = \frac{1.1949875 - 1.5374383}{\sqrt{1.338}} = -0.2960534513.$$

Die standardisierte Teststatistik besitzt nun die Form $T^* \sim \mathcal{N}(0, 1)$. $|T^*|$ besitzt daher eine Halbnormalverteilung mit Verteilungsfunktion F. Folglich lässt sich der

p-Wert ermitteln mit:

$$\begin{aligned} p(|T(b)^*|) &= p(|-0.2960534513|) = P(|T^*| \geq |-0.2960534513|) \\ &= 1 - P(|T^*| \leq 0.2960534513) \\ &= 1 - F(0.2960534513) \\ &= 1 - \frac{2}{\sqrt{\pi}} \int_0^{\frac{0.2960534513}{\sigma\sqrt{2}}} \exp(-t^2)\, dt \\ &= 0.767189. \end{aligned}$$

Der Vergleich von p-Wert und Signifikanzniveau α ergibt anschließend:

$$\alpha = 0.01 \leq 0.767189 = p(|T(b)^*|).$$

Das Testergebnis ist daher eine Entscheidung für H_0. Die Aussage des Tests ist, dass die betrachtete Bitsequenz b keinen Widerspruch zur anfangs formulierten Eigenschaft E geliefert hat. Daraus folgt also auch kein Widerspruch zur Zufälligkeit der Bitsequenz und des zugrundeliegenden Zufallszahlengenerators.

4.3 Cumulative Sums (Cusum) Test

4.3.1 Zweck des Tests

Der „Cumulative Sums (Cusum) Test“ legt den Fokus auf die Reihenfolge der verschiedenen Bits in der Sequenz. Eine Sequenz der Form 000...000111...111 mit gleich vielen 0- und 1-Bits würde den „Frequency Monobit Test“ bestehen. Ganz offensichtlich ist diese Sequenz jedoch nicht zufällig, da die Reihenfolge der Bits einem klar erkennbaren Prinzip folgt, in welchem zuerst nur 0-Bits und anschließend nur 1-Bits auftreten. Um ein angemessenes Maß für die Zufälligkeit der Reihenfolge von Bits und somit eine Teststatistik T zu finden, führen wir ein vereinfachtes Konzept des (eindimensionalen) *Random Walks* (Irrfahrt) ein:

Definition 4.3.1 (Eindimensionaler Random Walk (Irrfahrt))**.** Sei $X = (X_1, X_2, ...)$ eine Folge identisch verteilter Zufallsvariablen mit $X_1 \in \{-1, 1\}$. Die Folge $S = (S_0, S_1, ...)$, mit $S_0 = 0$ und

$$S_n = \sum_{k=1}^{n} X_k$$

ist ein *Random Walk* mit den *Schritten* $X_1, X_2, ...$ und der *Abweichung* S_n *zum Zeitpunkt* n[4, S. 165].

Um dieses Konzept auf eine Zahlenfolge b anwenden zu können, formt man wie in Kapitel 4.1.2 die identisch bernoulliverteilten Bits b_i in Zufallsvariablen $2b_i - 1 = X_i : \{0, 1\} \to \{-1, 1\}$ um, mit:

$$P(X_i = k) = \begin{cases} -1, & \text{für } k = 0 \\ 1, & \text{für } k = 1. \end{cases}$$

Man definiert die Folge $X = (X_1, X_2...)$ als die Schritte eines Random Walks S. Auf diese Weise kann nun für beliebige Zahlenfolgen b ein Random Walk S erstellt werden. Sei etwa $b = 1011010110$, so ist der entstehende Random Walk $S = (1, (-1), 1, 1, (-1), 1, (-1), 1, 1, 1)$, siehe Abbildung 6.

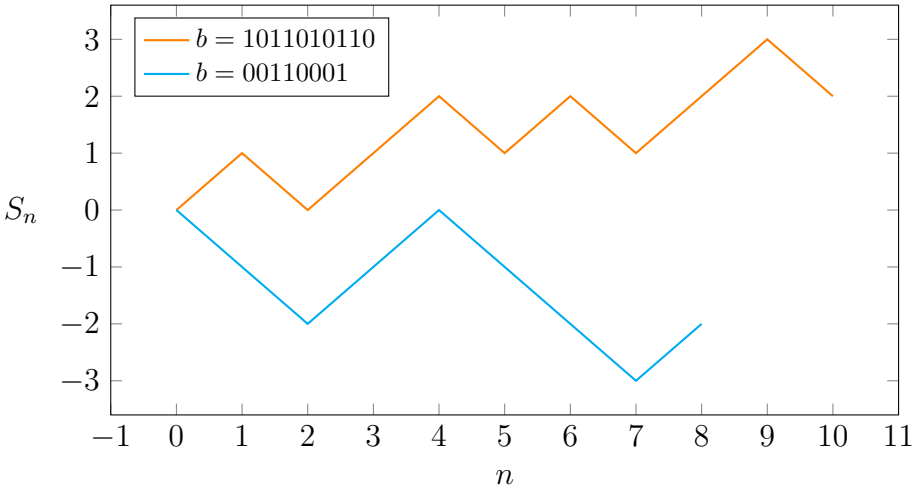


Abbildung 6: Abweichungen von eindimensionalen Random Walks für zwei Zahlenfolgen

Die Annahme des „Cumulative Sums (Cusum) Tests“ ist, dass zufällige Zahlenfolgen Random Walks erzeugen, deren maximale absolute Abweichung nah an 0 liegt. Eine Sequenz der Form 000...000111...111 besäße eine große maximale absolute Abweichung von 0, wodurch der Test H_0 ablehnen würde. Die zu testende Eigenschaft E ist: „Die maximale absolute Abweichung des aus b erzeugten Random Walks S liegt in einem für Zufallszahlen erwarteten Bereich.“ In der „NIST Statistical Test Suite“ gibt es zusätzlich die Option, den Random Walk rückwärts zu durchlaufen. Wir beschränken uns in der Testbeschreibung auf das Durchlaufen in Vorwärtsrichtung. Um den Random Walk rückwärts zu durchlaufen, kehrt man die Sequenz X der Schritte um und führt anschließend das gleiche Verfahren durch, das im Folgenden beschrieben wird.

4.3.2 Testbeschreibung

Wie bereits erwähnt, muss die maximale absolute Abweichung des von b erzeugten Random Walks in einem vorgegebenen Bereich liegen, damit der Test H_0 akzeptiert. Man definiert daher die Teststatistik mit:

$$T(b) = \max_{1 \leq k \leq n} |S_k|.$$

Für die Beispielsequenz $b = 1011010110$ ergibt sich die Teststatistik, wie zu erkennen aus Abbildung 6:

$$T(b) = \max_{1 \leq k \leq n} |S_k| = |S_9| = 3.$$

Die Verteilung von $\max_{1 \leq k \leq n} |S_k|$ wird in dieser Arbeit nicht genauer betrachtet. Wir halten uns daher an Satz 2.35. aus „Irrfahrten - Faszination der Random Walks“ von Norbert Henze[5, S. 95], welcher besagt:

Satz 4.1 (Verteilung des Betragsmaximums von Random Walks)**.** Sei $|M|_n = \max_{1 \leq k \leq n} |S_k|$ das Betragsmaximum des Random Walks S. Es gilt für jedes $k = 1, ..., n$:

$$\begin{aligned} P(|M|_n \geq k) &= 2 \cdot \sum_{s=1}^{\lfloor \frac{n}{2k} + \frac{1}{2} \rfloor} (-1)^{s-1} P(|M|_n \geq (2s-1)k) \\ &= 2 \cdot \sum_{s=1}^{\lfloor \frac{n}{2k} + \frac{1}{2} \rfloor} (-1)^{s-1} \sum_{j=(2s-1)k}^{n} \frac{1}{2^n} \binom{n}{\lfloor \frac{n+j+1}{2} \rfloor}. \end{aligned}$$

Mit der Aussage des Satzes 4.1 lässt sich nun der p-Wert für die berechnete Teststatistik $T(b) = |M|_n = 3$, mit $n = 10$ bestimmen:.

$$\begin{aligned} p(T(b)) &= p(3) = P(T(B) \geq 3) \\ &= 2 \cdot \sum_{s=1}^{\lfloor \frac{n}{2k} + \frac{1}{2} \rfloor} (-1)^{s-1} \sum_{j=(2s-1)k}^{n} \frac{1}{2^n} \binom{n}{\lfloor \frac{n+j+1}{2} \rfloor} \\ &= 2 \cdot \sum_{s=1}^{\lfloor \frac{10}{2 \cdot 3} + \frac{1}{2} \rfloor} (-1)^{s-1} \sum_{j=(2s-1) \cdot 3}^{10} \frac{1}{2^{10}} \binom{10}{\lfloor \frac{10+j+1}{2} \rfloor} \\ &= 2 \cdot \sum_{s=1}^{2} (-1)^{s-1} \sum_{j=(2s-1) \cdot 3}^{10} \frac{1}{2^{10}} \binom{10}{\lfloor \frac{11+j}{2} \rfloor} \\ &= 0.68359375. \end{aligned}$$

Der Vergleich von p-Wert und Signifikanzniveau α ergibt anschließend:

$$\alpha = 0.01 \leq 0.68359375 = p(T(b)).$$

Das Testergebnis ist daher eine Entscheidung für H_0. Die Aussage des Tests ist, dass die betrachtete Bitsequenz b keinen Widerspruch zur anfangs formulierten Eigenschaft E geliefert hat. Daraus folgt also auch kein Widerspruch zur Zufälligkeit der Bitsequenz und des zugrundeliegenden Zufallszahlengenerators.

5 Statistisches Testen eines Generators

In diesem Kapitel wird der Micali-Schnorr Generator mit den 15 Tests der NIST Statistical Test Suite getestet. Die gewählten Parameter werden begründet und die ausgegebenen Ergebnisse werden interpretiert. Der getestete Generator hat im Allgemeinen keinen Einfluss auf die Art und Weise des Testhergangs, da ausschließlich die ausgegebenen Zahlenfolgen untersucht werden sollen. Jeder einzelne statistische Test kann nur eine Eigenschaft einer Zahlenfolge testen. Deshalb sollten mehrere verschiedene Tests durchgeführt werden, damit eine Zahlenfolge als zufällig eingestuft werden kann. Um letztendlich den zugrundeliegenden Bitgenerator als PRNG bezeichnen zu können, muss jedoch mehr als nur eine Ausgabe des PRNGs getestet werden. Es werden also mehrere Ausgaben eines Generators mit jeweils mehreren Tests auf verschiedene Eigenschaften getestet. Hier stellen sich zwei Fragen: Wie viele Tests müssen pro Zahlenfolge durchgeführt werden und wie viele Zahlenfolgen müssen pro Generator getestet werden, damit man sicher sein kann, dass der Bitgenerator pseudozufällige Ausgaben generiert?

5.1 Parameterwahl

Die Anzahl der zu testenden Zahlenfolgen pro Generator hängt vom gewählten Signifikanzniveau der Tests ab. Ein Signifikanzniveau $\alpha = 0.01$ sorgt im Fall einer geltenden Nullhypothese dafür, dass erwartungsgemäß einer aus 100 Tests zur Ablehnung von H_0 führt. Entsprechend wäre es nicht ausreichend, weniger als 100 Zahlenfolgen zu testen. Die Anzahl m der zu testenden Zahlenfolgen kann also mit $m \geq \frac{1}{\alpha}$ nach unten beschränkt werden. Wir wählen für unsere Tests das standardmäßige Niveau des NIST Testsets $\alpha = 0.01$, wodurch sich eine Mindestanzahl von 100 Zahlenfolgen ergibt. Wir nutzen $m = 1000$ Zahlenfolgen. Zu erwarten sind also etwa 10 Tests, die zur Ablehnung von H_0 führen, wenn H_0 tatsächlich gilt.

Eine ausreichende Anzahl Tests für einen Generator gibt es nicht. Der Informatiker Andrew C. Yao formulierte den Satz: Ein Bitgenerator ist genau dann ein Pseudozufallszahlengenerator, wenn er alle statistischen Tests besteht[16, S. 84]. Da es nicht möglich ist alle statistischen Tests auf einen Bitgenerator anzuwenden, wählt man sich stattdessen eine Menge an Tests aus, die man als ausreichend betrachtet. Hierbei kann man auf bereits zusammengestellte, online verfügbare Testsets und ihre Implementierungen zurückgreifen. Weit verbreitete Testsets sind zum Beispiel „TestU01“ von L'ecuyer und Simard[12] oder die Tests der NIST Statistical Test Suite, auf die in Kapitel 4 eingegangen wurde. Das Testen der verschiedenen Zahlenfolgen mit mehreren Tests liefert eine große Menge an einzelnen Testergebnissen. Von diesen möchte man auf die Zufälligkeit des zugrundeliegenden

Bitgenerators schließen. Entsprechend muss man die vielen erhaltenen Testergebnisse bündeln, um ein finales Gesamtergebnis zu erhalten.

Neben der Anzahl der zu testenden Zahlenfolgen und der verwendeten Tests gibt es noch weitere Parameter, die für verschiedene Tests unterschiedliche Werte besitzen sollten. Der grundlegendste hierbei ist die Länge n der Zahlenfolgen für einen Test. Während der Frequency Test oder der CuSum Test bereits für Zahlenfolgen der Länge $n \geq 100$ verwendbare Ergebnisse liefert, benötigt der Universal Statistical Test von Maurer in der Implementierung des NIST Testsets $n \geq 387840$ Bits, um die Sequenz angemessen zu unterteilen und zu testen. Für einige in dieser Arbeit nicht erwähnte Tests des NIST Testsets gilt ein Mindestwert von $n = 1000000$, weshalb wir diesen für unseren Test nutzen.

Einige Tests teilen die Zahlenfolge in Blöcke auf und führen dann blockweise Operationen durch oder arbeiten mit den Mustern, die in den Blöcken auftreten, um eine Teststatistik zu berechnen. Hier passt man grundsätzlich die Größe dieser Blöcke an die Längen der Zahlenfolgen an. Für kurz gewählte Zahlenfolgen und lang gewählte Blöcke kann es zum Beispiel passieren, dass manche Muster in keinem Block auftreten, da nur eine begrenzte Anzahl an Blöcken aus der Zahlenfolge gebildet werden kann. Dadurch kann meist keine sinnvolle Teststatistik gebildet werden und das Ergebnis kann fehlerhaft sein. Der Parameter L von Maurers Test wird daher zum Beispiel in der Implementierung vom NIST in Abhängigkeit von der Länge der Zahlenfolge gewählt. Für unsere gewählte Länge von $n = 1000000$ arbeitet der Test mit einer Blockgröße von $L = 7$. Für alle weiteren Tests, die mit Blockeinteilungen arbeiten verwenden wir die standardmäßig vorgegebenen Parameter der NIST Implementierung (siehe Anhang A Tabelle 7).

5.2 Interpretation der Testergebnisse

Nach Ausführen der NIST Implementierung der Tests erhält man Tabelle 6 aus Anhang A, in der man jeden Test in einer eigenen Zeile findet. Wir betrachten ausgewählte Zeilen in Tabelle 4. Für das Bilden eines Gesamtergebnisses aus allen ermittelten Testergebnissen betrachtet man die 15 verschiedenen Tests separat. Jeder dieser Tests besitzt eine sogenannte *Testergebnismenge*, in welcher sich die Ergebnisse der Tests von m Zahlenfolgen befinden. Man schaut sich nun die Ergebnismenge eines Tests an und entscheidet mithilfe dieser Menge, ob der Generator den zugehörigen Test bestanden hat. Dies führt man für alle 15 Tests durch und erhält dadurch 15 *Teilergebnisse*. Spricht eines der Teilergebnisse gegen die Zufälligkeit des getesteten Generators, so gilt dieser nicht als PRNG. Um von einer Testergebnismenge auf ein Teilergebnis zu schließen, müssen zwei Kriterien betrachtet werden.

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	p_a	Bestandene Tests	Test
88	97	111	99	105	113	105	90	101	91	0.662091	988/1000	Frequency
97	96	99	104	100	109	108	104	89	94	0.935716	989/1000	CuSum
94	83	123	118	96	90	99	103	101	93	0.139655	985/1000	CuSum
134	96	91	93	86	104	112	95	111	78	0.006472	982/1000	Maurers Test

Tabelle 4: Ausgewählte Testergebnisse des Micali-Schnorr Generators. Für den CuSum Test ergeben sich zwei Teilergebnisse für das Betrachten der Zahlenfolgen in Vorwärts- beziehungsweise Rückwärtsrichtung

5.2.1 Anteil bestandener Tests

Das NIST legt für jede Testergebnismenge ein Intervall fest, in dem der Anteil der Testergebnisse mit $f(b) = 1$ liegen muss, damit der zugehörige Test als bestanden gilt. Dieses Intervall beschreibt sich durch: $\hat{p} \pm 3\sqrt{\frac{\hat{p}(1-\hat{p})}{m}}$, mit $\hat{p} = 1-\alpha$. In unserem Fall, mit $m = 1000$ Zahlenfolgen und $\hat{p} = 1 - 0.01 = 0.99$, lautet das Intervall:

$$[0.980560, 0.999439] \approx [981/1000, 999/1000].$$

Folglich führen Testergebnismengen mit 1000 oder ≤ 980 Ergebnissen $f(b) = 1$ zu einem Teilergebnis, das gegen die Zufälligkeit des Generators spricht. Dies begründet sich auf der Wahrscheinlichkeit eines Fehlers erster Art von $\alpha = 0.01$, wonach wie bereits erwähnt etwa 10 Tests nicht bestanden werden, wenn H_0 gilt. Das Intervall stellt also einen Bereich um diese erwarteten ablehnenden 10 Testergebnisse $f(b) = 0$ dar, indem man für die bestandenen Tests mit $f(b) = 1$ einen Bereich um den erwarteten Wert 990 festlegt. In Tabelle 4 lässt sich erkennen, dass alle Teilergebnisse einen Anteil bestandener Tests aufweisen, der in dem geforderten Intervall liegt. Das beschriebene Kriterium trifft also für den Micali-Schnorr Generator zu, weshalb wir keine Aussage gegen die Zufälligkeit des Generators treffen können.

5.2.2 Gleichverteilung der p-Werte

Neben der Testergebnismenge werden außerdem noch die p-Werte jedes einzelnen Tests einer Zahlenfolge betrachtet. Für jeden der 15 Tests werden m p-Werte auf ihre Gleichverteilung untersucht. Denn wenn die berechneten p-Werte (nicht)

gleichverteilt sind, dann sind auch die getesteten Zahlenfolgen (nicht) gleichverteilt, wodurch der entsprechende Generator als (nicht) zufällig zu betrachten wäre. Diesen Zusammenhang

$$\text{Gleichverteilung der } p\text{-Werte} \Leftrightarrow \text{Gleichverteilung der Zahlenfolgen } b$$

erläutern wir anhand eines Beispiels mit einer standardnormalverteilten Teststatistik T.

Beispiel 5.1. Sei $T(b)$ eine Instanz der Teststatistik $T \sim \mathcal{N}(0,1)$. Weiterhin sei D eine Zufallsvariable, die die Instanzen einer standardnormalverteilten Zufallsvariablen abbildet auf die Intervalle D_i, mit $1 \leq i \leq 10$ und der Gleichverteilung

$$P(D = D_j) = \int_{D_j} \varphi(x)\, dx = 0.1, \text{ für } j \in \{1, 2, ..., 10\}, \text{ siehe Abbildung 7.}$$

Damit D tatsächlich eine Gleichverteilung besitzt, muss die zugrundeliegende Teststatistik T auch tatsächlich eine Standardnormalverteilung besitzen, was wiederum voraussetzt, dass die getesteten Zahlenfolgen gleichverteilt sind.

$$\begin{aligned} \text{Die Zahlenfolgen } b \text{ sind gleichverteilt.} &\Leftrightarrow T(b) \text{ ist standardnormalverteilt.} \\ &\Leftrightarrow D \text{ ist gleichverteilt.} \\ &\Leftrightarrow C \text{ ist gleichverteilt.} \end{aligned}$$

C beschreibt hierbei die Bildung des p-Wertes $p(T(b))$ als bijektive Abbildung von den Intervallen D_i auf Intervalle $C_i = \left[\frac{i-1}{10}, \frac{i}{10}\right]$. Die Aussagen

$$T(b) \in D_j \qquad \text{und} \qquad p(T(b)) \in C_j$$

sind äquivalent.

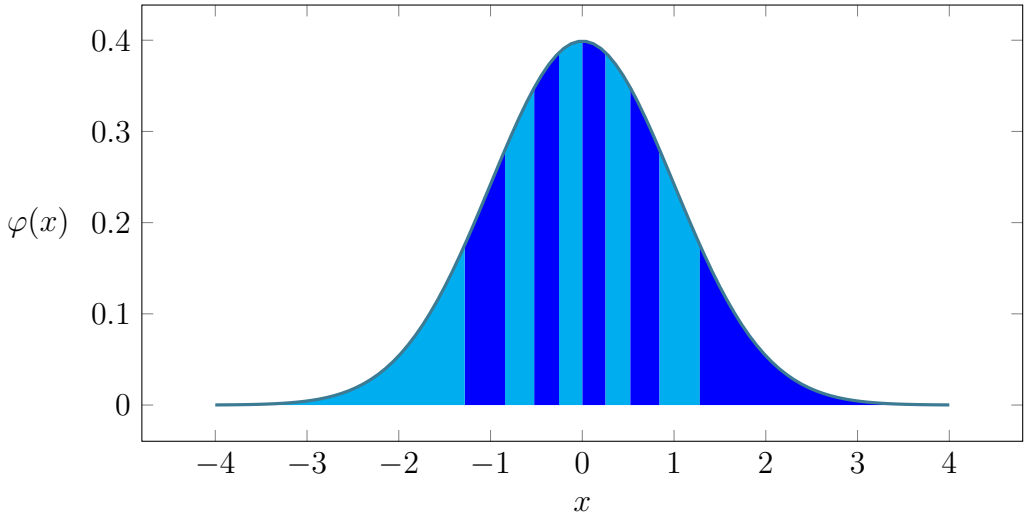


Abbildung 7: Standardnormalverteilung mit eingefärbten Bereichen D_i mit $P(D_i) = \int_{D_i} \varphi(x)\,dx = 0.1$ und $1 \leq i \leq 10$

In Tabelle 4 beschreiben die ersten zehn Spalten C_1 bis C_{10} die Anzahl der p-Werte in dem Intervall C_i. Für den Frequency Test können diese Werte wie in Abbildung 8 dargestellt werden. Bereits durch das Betrachten des Balkendiagramms kann eine Gleichverteilung der p-Werte vermutet werden.

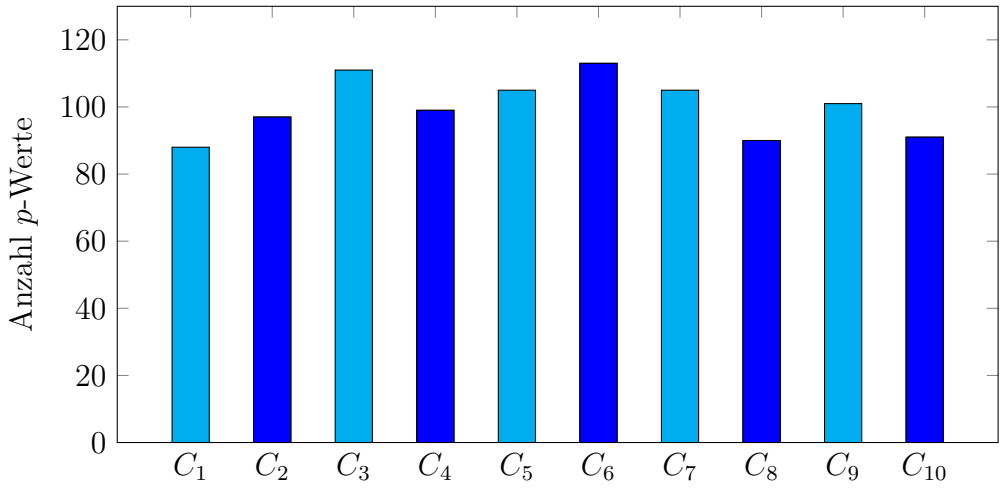


Abbildung 8: Aufteilung der p-Werte auf die Intervalle $C_i = \left[\frac{i-1}{10}, \frac{i}{10}\right]$

Um mit diesen Werten eine verlässliche Aussage über die Verteilung der p-Werte zu treffen, wird ein sogenannter *Anpassungstest* durchgeführt[7, S. 250]. Mit

einem solchen Test lässt sich durch den Wert des dort ermittelten p-Wertes p_a eine Aussage darüber treffen, ob die tatsächliche Verteilung der p-Werte der anfangs vermuteten Gleichverteilung entspricht. Auf die Details dieses Tests wird in der NIST Test Suite und in dieser Arbeit nicht genauer eingegangen. Das Ergebnis des Tests ist ein p-Wert p_a, der laut NIST die Bedingung $p_a \geq 0.0001$ erfüllen muss, damit den p-Werten eine Gleichverteilung zugeschrieben werden kann. Dieser findet sich ebenfalls in der Tabelle 4.

Wie zu erkennen, gilt für alle dort ermittelten Werte $p_a \geq 0.0001$. Daraus lässt sich die Gleichverteilung der getesteten Zahlenfolgen nicht widerlegen und auch dieses Kriterium wird vom Micali-Schnorr Generator erfüllt. Der Micali-Schnorr Generator hat nach Betrachtung beider Kriterien zur Interpretation von Testergebnissen keinen Widerspruch zur Zufälligkeit seiner Ausgaben geliefert. Entsprechend bestätigt auch dieser Testdurchlauf, dass es sich bei diesem Generator um einen Pseudozufallszahlengenerator handelt.

6 Fazit

Diese Arbeit sollte die Grundlagen des statistischen Testens vermitteln und dessen Rolle in Hinblick auf den Anwendungsbereich Kryptographie beschreiben. Das Testset des NIST sollte mit den Grundlagen in Verbindung gebracht werden. Ein Pseudozufallszahlengenerator sollte mit diesem Testset untersucht werden. Ziel war es, das generelle Testen von Zufallszahlengeneratoren anhand des gezeigten Testablaufs zu erläutern und die Testergebnisse zu interpretieren.

Wir haben die stochastischen Grundlagen für das statistische Testen erläutert. Hierbei haben wir verschiedene Verteilungen erklärt, wie etwa Bernoulli-Verteilung, Binomialverteilung, Gleichverteilung und verschiedene Formen der Normalverteilung. Die Anwendung der Zentralen Grenzwertsätze in der Kryptographie wurde beschrieben. Mithilfe von Beispielen sind Konzepte, wie Testproblem oder Signifikanztests mit kritischen Bereichen und p-Werten, eingeführt worden. Weiterhin haben wir diverse Zufallszahlengeneratoren und andere grundlegende Begriffe der Kryptographie definiert. Nicht behandelt wurden die Chi-Quadrat-Verteilung und die zugehörigen Tests auf Anpassung, Unabhängigkeit und Homogenität.

Statistische Tests dienen der Überprüfung von Zufallszahlen. Diese werden zur Verschlüsselung und zum Signieren von Daten genutzt. Eine zufällige Zahlenfolge kann mit den Eigenschaften eines mehrmaligen Münzwurfs beschrieben werden. Sie ist gleichverteilt mit der Länge n über die Menge $\{bin(0), bin(1), ..., bin(2^n)\}$ und besteht aus stochastisch unabhängigen, identisch bernoulli-verteilten Bits. Um diese Eigenschaften nachzuweisen, verwendet das Testset des NIST 15 statistische Tests. Der Frequency Test überprüft, ob Zahlenfolgen in etwa die gleiche Anzahl 0-Bits und 1-Bits besitzen. Maurers Test stellt fest, ob eine Bitsequenz nicht komprimierbar ist. Der CuSum Test überprüft, ob das Betragsmaximum des von der Zahlenfolge erzeugten Random Walks in einem vorbestimmten Bereich liegt. Alle drei Tests bestätigen die Nullhypothese H_0, wenn die jeweils getestete Eigenschaft nicht widerlegt werden kann.

Dem statistischen Testen eines Zufallszahlengenerators geht eine sinnvolle Wahl der Parameter voraus. Hierbei muss die Anzahl m der zu testenden Zahlenfolgen in Abhängigkeit vom Signifikanzniveau α gewählt werden, mit $m \geq \frac{1}{\alpha}$. Die zu nutzenden Tests müssen ausgewählt werden, wobei auf bereits zusammengestellte Testsets zurückgegriffen werden kann. Die Länge der einzelnen Sequenzen und eventuelle Blockgrößen müssen in Abhängigkeit von den Anforderungen des jeweiligen Tests gewählt werden. Die Interpretation der Testergebnisse erfolgt durch das Überprüfen des Anteils an bestandenen Tests und der Gleichverteilung der p-Werte. Für den Micali-Schnorr Generator konnte mithilfe der ermittelten Testergebnisse kein Widerspruch zu seiner Pseudozufälligkeit gefunden werden.

Literatur

[1] Lawrence E Bassham III u. a. *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology, 2010.

[2] William Feller. "On the normal approximation to the binomial distribution". In: *Selected Papers I*. Springer, 2015, S. 655–665.

[3] Dirk Fox. "Fälschungssicherheit digitaler Signaturen". In: *Datenschutz und Datensicherheit (DuD)* 2.97 (1997), S. 69–74.

[4] Bert E Fristedt und Lawrence F Gray. *A modern approach to probability theory*. Springer Science & Business Media, 2013.

[5] Norbert Henze. *Irrfahrten und verwandte Zufälle*. Springer.

[6] Norbert Henze. *Stochastik für Einsteiger*. Springer Fachmedien Wiesbaden, 2018. DOI: 10.1007/978-3-658-22044-0. URL: https://doi.org/10.1007/978-3-658-22044-0.

[7] Perry R Hinton. *Statistics explained*. Routledge, 2014.

[8] Dudenredaktion (o. J.) *„Zufall" auf Duden online*. https://www.duden.de/rechtschreibung/Zufall. [Online; letzter Zugriff: 20. April 2020]. 2020.

[9] Jonathan Katz und Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2014.

[10] Cameron F Kerry und Patrick D Gallagher. "Digital signature standard (DSS)". In: *FIPS PUB* (2013), S. 186–4.

[11] Andrei Nikolaevich Kolmogorov. "Three approaches to the quantitative definition of information". In: *International journal of computer mathematics* 2.1-4 (1968), S. 157–168.

[12] Pierre L'Ecuyer und Richard Simard. "TestU01: AC library for empirical testing of random number generators". In: *ACM Transactions on Mathematical Software (TOMS)* 33.4 (2007), S. 1–40.

[13] FC Leone, LS Nelson und RB Nottingham. *The Folded Normal Distribution*. Mathematical Sciences Directorate, Air Force Office of Scientific Research, 1961.

[14] Ueli M Maurer. "A universal statistical test for random bit generators". In: *Journal of cryptology* 5.2 (1992), S. 89–105.

[15] Alfred Menezes, Paul C. van Oorschot und Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN: 0-8493-8523-7. DOI: 10.1201/9781439821916. URL: http://cacr.uwaterloo.ca/hac/.

[16] Andrew C Yao. “Theory and application of trapdoor functions”. In: *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*. IEEE. 1982, S. 80–91.

A Anhang

z	0	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0	0,5	0,50399	0,50798	0,51197	0,51595	0,51994	0,52392	0,5279	0,53188	0,53586
0,1	0,53983	0,5438	0,54776	0,55172	0,55567	0,55962	0,56356	0,56749	0,57142	0,57535
0,2	0,57926	0,58317	0,58706	0,59095	0,59483	0,59871	0,60257	0,60642	0,61026	0,61409
0,3	0,61791	0,62172	0,62552	0,6293	0,63307	0,63683	0,64058	0,64431	0,64803	0,65173
0,4	0,65542	0,6591	0,66276	0,6664	0,67003	0,67364	0,67724	0,68082	0,68439	0,68793
0,5	0,69146	0,69497	0,69847	0,70194	0,7054	0,70884	0,71226	0,71566	0,71904	0,7224
0,6	0,72575	0,72907	0,73237	0,73565	0,73891	0,74215	0,74537	0,74857	0,75175	0,7549
0,7	0,75804	0,76115	0,76424	0,7673	0,77035	0,77337	0,77637	0,77935	0,7823	0,78524
0,8	0,78814	0,79103	0,79389	0,79673	0,79955	0,80234	0,80511	0,80785	0,81057	0,81327
0,9	0,81594	0,81859	0,82121	0,82381	0,82639	0,82894	0,83147	0,83398	0,83646	0,83891
1	0,84134	0,84375	0,84614	0,84849	0,85083	0,85314	0,85543	0,85769	0,85993	0,86214
1,1	0,86433	0,8665	0,86864	0,87076	0,87286	0,87493	0,87698	0,879	0,881	0,88298
1,2	0,88493	0,88686	0,88877	0,89065	0,89251	0,89435	0,89617	0,89796	0,89973	0,90147
1,3	0,9032	0,9049	0,90658	0,90824	0,90988	0,91149	0,91309	0,91466	0,91621	0,91774
1,4	0,91924	0,92073	0,9222	0,92364	0,92507	0,92647	0,92785	0,92922	0,93056	0,93189
1,5	0,93319	0,93448	0,93574	0,93699	0,93822	0,93943	0,94062	0,94179	0,94295	0,94408
1,6	0,9452	0,9463	0,94738	0,94845	0,9495	0,95053	0,95154	0,95254	0,95352	0,95449
1,7	0,95543	0,95637	0,95728	0,95818	0,95907	0,95994	0,9608	0,96164	0,96246	0,96327
1,8	0,96407	0,96485	0,96562	0,96638	0,96712	0,96784	0,96856	0,96926	0,96995	0,97062
1,9	0,97128	0,97193	0,97257	0,9732	0,97381	0,97441	0,975	0,97558	0,97615	0,9767
2	0,97725	0,97778	0,97831	0,97882	0,97932	0,97982	0,9803	0,98077	0,98124	0,98169
2,1	0,98214	0,98257	0,983	0,98341	0,98382	0,98422	0,98461	0,985	0,98537	0,98574
2,2	0,9861	0,98645	0,98679	0,98713	0,98745	0,98778	0,98809	0,9884	0,9887	0,98899
2,3	0,98928	0,98956	0,98983	0,9901	0,99036	0,99061	0,99086	0,99111	0,99134	0,99158
2,4	0,9918	0,99202	0,99224	0,99245	0,99266	0,99286	0,99305	0,99324	0,99343	0,99361
2,5	0,99379	0,99396	0,99413	0,9943	0,99446	0,99461	0,99477	0,99492	0,99506	0,9952
2,6	0,99534	0,99547	0,9956	0,99573	0,99585	0,99598	0,99609	0,99621	0,99632	0,99643
2,7	0,99653	0,99664	0,99674	0,99683	0,99693	0,99702	0,99711	0,9972	0,99728	0,99736
2,8	0,99744	0,99752	0,9976	0,99767	0,99774	0,99781	0,99788	0,99795	0,99801	0,99807
2,9	0,99813	0,99819	0,99825	0,99831	0,99836	0,99841	0,99846	0,99851	0,99856	0,99861
3	0,99865	0,99869	0,99874	0,99878	0,99882	0,99886	0,99889	0,99893	0,99896	0,999
3,1	0,99903	0,99906	0,9991	0,99913	0,99916	0,99918	0,99921	0,99924	0,99926	0,99929
3,2	0,99931	0,99934	0,99936	0,99938	0,9994	0,99942	0,99944	0,99946	0,99948	0,9995
3,3	0,99952	0,99953	0,99955	0,99957	0,99958	0,9996	0,99961	0,99962	0,99964	0,99965
3,4	0,99966	0,99968	0,99969	0,9997	0,99971	0,99972	0,99973	0,99974	0,99975	0,99976
3,5	0,99977	0,99978	0,99978	0,99979	0,9998	0,99981	0,99981	0,99982	0,99983	0,99983
3,6	0,99984	0,99985	0,99985	0,99986	0,99986	0,99987	0,99987	0,99988	0,99988	0,99989
3,7	0,99989	0,9999	0,9999	0,9999	0,99991	0,99991	0,99992	0,99992	0,99992	0,99992
3,8	0,99993	0,99993	0,99993	0,99994	0,99994	0,99994	0,99994	0,99995	0,99995	0,99995
3,9	0,99995	0,99995	0,99996	0,99996	0,99996	0,99996	0,99996	0,99996	0,99997	0,99997
4	0,99997	0,99997	0,99997	0,99997	0,99997	0,99997	0,99998	0,99998	0,99998	0,99998

Tabelle 5: Standardtabelle einer Standardnormalverteilung

C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	p_a	Bestandene Tests	Test
88	97	111	99	105	113	105	90	101	91	0.662091	988/1000	Frequency
102	100	87	99	97	99	97	118	107	94	0.737915	988/1000	Block Frequency
97	96	99	104	100	109	108	104	89	94	0.935716	989/1000	CuSum
94	83	123	118	96	90	99	103	101	93	0.139655	985/1000	CuSum
78	91	108	112	118	103	86	103	92	109	0.103753	996/1000	Runs
101	114	79	117	99	85	98	114	99	94	0.125927	993/1000	Longest Run
112	99	101	97	105	72	97	115	105	97	0.195864	994/1000	Matrix Rank
121	86	92	97	93	98	116	91	89	117	0.088226	995/1000	DFT
117	101	108	106	99	82	99	99	95	94	0.556460	982/1000	Overlapping Template
134	96	91	93	86	104	112	95	111	78	0.006472	982/1000	Maurers Test
98	97	101	107	100	104	97	98	104	94	0.997568	993/1000	Approximate Entropy
116	103	111	81	97	93	98	98	89	114	0.255705	986/1000	Serial
115	113	100	111	82	81	101	97	92	108	0.146152	991/1000	Serial
117	103	97	104	90	82	91	99	113	104	0.339271	987/1000	Linear Complexity

Tabelle 6: Testergebnisse des Micali-Schnorr Generators. Die Tests Non Overlapping Template, Random Excursion und Random Excursion Variant wurden aus Platzgründen nicht dargestellt.

Test	**Blockgröße in Bit**
Block Frequency	128
Non-overlapping Template	9
Overlapping Template	9
Approximate Entropy	10
Serial	16
Linear Complexity	500

Tabelle 7: Standardblockgrößen der NIST Tests mit Blockunterteilungen

B Selbstständigkeitserklärung

Hiermit erkläre ich, Julian Müller, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen Hilfsmittel als angegebene verwendet habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Werken entnommen sind, habe ich als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch bei keinem anderen Prüfer als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht

Lehrte, den 4. Mai 2020

Julian Müller