

Anwendungshinweise und Interpretationen zum Schema (AIS)

AIS 20, Version 1

Stand: 02.12.1999

Status: verbindlich

Thema: Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren

Herausgeber: Zertifizierungsstelle des BSI, Referat II2 im Rahmen des Zertifizierungsschemas

Verteiler: Lizenzierte Prüfstellen¹
Private Zertifizierungs- und Bestätigungsstellen²
BSI-intern: Referate II2, II4, V1, V3, III5

¹ Alle Evaluatoren in den vom BSI für Evaluierungen nach ITSEC oder CC lizenzierten Stellen

² Alle Zertifizierer der Privaten Stellen, die vom BSI anerkannte Deutsche IT-Sicherheitszertifikate oder Bestätigungen nach dem deutschen Signaturgesetz herausgeben

Dienstgebäude:

Nr. 1: Godesberger Allee 183
Bonn-Hochkreuz
Tel.: (0228) 9582-0
Fax: (0228) 9582-400

Nr. 2: Mainzer Straße 84
Bonn-Mehlem
Tel.: (0228) 9582-0
Fax: (0228) 9582-750

Nr. 3: Merianstraße 100
Köln-Chorweiler
Tel.: (0221) 97959-0
Fax: (0221) 97959-250

Nr. 4: Gabrielweg 5
Swisttal-Heimerzheim
Tel.: (02254) 9403-0
Fax: (02254) 9403-40

Nr. 5: Kessenicher Straße 216
Bonn-Dottendorf
Tel.: (0228) 9582-0
Fax: (0228) 9582-455

Kontoverbindung: Bundeskasse Bonn bei der Landeszentralbank Bonn (BLZ 380 000 00) Konto-Nr. 380 01 060 zugunsten des BSI

Internet: <http://www.bsi.bund.de/>

1 Hintergrund der AIS

Deterministische Zufallszahlengeneratoren sind in vielen Produkten enthalten. Teils werden sie zur Erzeugung einer Challenge als Teil eines Authentisierungsverfahrens oder zur Erzeugung von Signatur- oder Verschlüsselungsschlüsseln verwendet.

In bestimmten Fällen wie z.B. im Rahmen der Evaluierung einer Signaturkomponente, bei der Evaluierung von Authentisierungsverfahren oder bei der Evaluierung von Komponenten zur Schlüsselerzeugung ist eine Analyse und Bewertung des Zufallszahlengenerators notwendig.

Die Evaluationshandbücher liefern hierzu keine Information.

Eine weitgehend einheitliche Evaluierungsmethodik durch alle nach ITSEC oder CC evaluierenden, zertifizierenden oder bestätigenden Stellen ist notwendig.

Im nachfolgend vollständig zitierten Dokument „Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, Version 2.0 vom 02.12.1999“ wird ein Ansatz zur Beschreibung und Evaluierung formuliert.

Gegenüber der Entwurfsfassung (Version 1.5) vom 16.02.1999 wurde das Dokument unter Einbeziehung der Kommentierungen von Prüfstellen und Stellen im BSI in mehreren Schritten neu strukturiert und erweitert und in der jetzt vorliegenden Version 2.0 abgeschlossen.

Die AIS wird damit verbindlich.

Eine weitere Überarbeitung und Ergänzung kann zu einem späteren Zeitpunkt erfolgen, wenn weitergehende Erkenntnisse oder Erfahrungen aus der Anwendung vorliegen.

Priv.-Doz. Dr. Werner Schindler
BSI, III 5

Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren

Version 2.0
02.12.1999

Inhaltsverzeichnis

A. Motivation, Zielsetzung und inhaltliche Übersicht

A.1 Motivation und Zielsetzung: Zufallszahlen spielen in vielen kryptographischen Anwendungen eine wichtige Rolle. Dennoch gibt es bislang keine einheitlichen Evaluationskriterien für Zufallszahlengeneratoren. In diesem Papier werden Evaluationskriterien für deterministische Zufallszahlengeneratoren vorgestellt. Der Grundgedanke besteht darin, die Eignung deterministischer Zufallszahlengeneratoren in Abhängigkeit von den kryptographischen Anwendungen zu beurteilen, in denen sie eingesetzt werden.

A.2 Inhaltliche Übersicht: In Kapitel B wird der Untersuchungsgegenstand beschrieben. In Kapitel C werden vier Funktionalitätsklassen (K1, K2, K3, K4) eingeführt. Diese Einteilung wird begründet. Kapitel D beschreibt die Aufgaben des Evaluators. Die praktische Anwendbarkeit der Funktionalitätsklassen und die Aufgaben des Evaluators werden in Kapitel E an mehreren Beispielen erläutert.

A.3 Bemerkung: Die im folgenden definierten Funktionalitätsklassen beschreiben hierarchische Anforderungen, dies jedoch nicht auf dem Abstraktionsniveau der generischen Oberbegriffe der ITSEC, sondern auf der Spezifikationsebene technischer Eigenschaften.

Wird vom Antragsteller ein deterministischer Zufallszahlengenerator verwendet, der keiner der Funktionalitätsklassen K1 bis K4 zugeordnet werden kann und wird ein Deutsches IT-Sicherheitszertifikat angestrebt, ist eine Abstimmung mit dem BSI erforderlich.

B. Definitionen und Notation

B.1 Mathematische Beschreibung: Ein deterministischer Zufallszahlengenerator erzeugt auf deterministische Weise Zufallszahlen, die einzig und allein vom Anfangszustand („seed“) abhängen. Das 5-Tupel (S, R, ϕ, ψ, p_A) beschreibt die logische Struktur des Generators und der Seedauswahl. Im einzelnen sind

S	die (endliche) Menge der möglichen inneren Zustände des Zufallszahlengenerators,
R	die Menge der möglichen Ausgabewerte (Zufallszahlen),
$\phi: S \rightarrow S$	die Zustandsfunktion,
$\psi: S \rightarrow R$	die Ausgabefunktion,
p_A	ein Wahrscheinlichkeitsmaß, das die zufällige Verteilung des Anfangszustands $s_0 \in S$ beschreibt.

In Schritt $n \geq 1$ wird zunächst der innere Zustand mittels $s_n := \phi(s_{n-1}) \in S$ erneuert und danach eine Zufallszahl $r_n := \psi(s_n) \in R$ berechnet und ausgegeben.

B.2 Bemerkung: Obwohl der Auswahlmechanismus des Anfangszustands strenggenommen nicht zur Beschreibung eines deterministischen Zufallszahlengenerators gehört, ist seine Wahrscheinlichkeitsverteilung für eine Bewertung der zu erwartenden Zufallszahlenfolgen keineswegs unwesentlich.

B.3 Notation und Sprachgebrauch: Anstelle von „deterministischer Zufallszahlengenerator“ schreiben wir im folgenden kürzer „DRNG“ (deterministic random number generator). Ist im folgenden von DRNG die Rede, ist damit nicht dessen technische Realisierung gemeint, sondern das beschreibende 5-Tupel (S, R, ϕ, ψ, p_A) .

C. Funktionalitätsklassen

C.0 Motivation zur Einführung von Funktionalitätsklassen: Anders als bei physikalischen Rauschquellen kann ein deterministischer Zufallszahlengenerator durch das Erzeugen neuer Zufallszahlen die Gesamtentropie der Zufallszahlenfolge nicht über die Entropie des seed hinaus erhöhen. Folgen deterministisch erzeugter Zufallszahlen können daher nicht „zufällig“ im eigentlichen Sinn sein, sondern sie können sich bestenfalls im Hinblick auf bestimmte Eigenschaften ähnlich wie „echte“, d.h. wirklich zufällige Folgen verhalten.

Zur Charakterisierung „guter“ Zufallszahlenfolgen sind in der Literatur verschiedenste Versuche unternommen worden. Im Kontext kryptographischer Anwendungen verdienen vor allem [FI140] (4.11.1) und [IEP] (G.4.5) Erwähnung. Während die erste Quelle statistische Tests formuliert, macht die zweite die Eignung eines deterministischen Zufallszahlengenerators an der praktischen Unvorhersagbarkeit der von ihm erzeugten Zufallszahlenfolgen fest. Es sei bemerkt, dass der statistische Ansatz zur Bewertung von Pseudozufallszahlen in den beiden letzten Jahrzehnten intensiv im Zusammenhang mit stochastischen Simulationen verfolgt worden ist. Allerdings werden dort im allgemeinen andere Anforderungen an die Zufallszahlen gestellt als bei kryptographischen Anwendungen (siehe auch Kapitel C, K2.e)).

Da aber auch verschiedene kryptographische Anwendungen unterschiedliche Anforderungen an die benötigten Zufallszahlen stellen und deterministisch erzeugte Zufallszahlen sich ohnehin nur in bezug auf bestimmte Kriterien wie „echte“ Zufallszahlen verhalten können, legt dies den Gedanken nahe, die Eignung deterministischer Zufallszahlengeneratoren in Abhängigkeit von der bzw. den avisierten Anwendungen zu bewerten. Nachfolgend werden vier abwärtskompatible Funktionalitätsklassen (K1, K2, K3, K4) definiert und ausführlich besprochen. Die praktische Anwendbarkeit der Kriterien wird in Kapitel E an sechs Beispielen demonstriert.

C.1 Vom Antragsteller anzugeben:

- (i) Angabe der angestrebten Funktionalitätsklasse (K1, K2, K3, K4) mit Mechanismenstärke
- (ii.a) vollständige und nachvollziehbare informelle Beschreibung des deterministischen Zufallszahlengenerators.

- (ii.b) beschreibendes 5-Tupel (S, R, ϕ, ψ, p_A)
- (iii) eine obere Schranke M für die Anzahl der Zufallszahlen, die mit dem DRNG während seines gesamten Lebenszyklus bzw. bis zu einem Neustart mit einem neuem, gemäß p_A ausgewählten Anfangszustand $s_0 \in S$ maximal generiert werden
- (iv) nachvollziehbare Beschreibung, wie der seed generiert wird samt Begründung, weshalb auf diese Weise die Verteilung p_A induziert werden soll.
- + zusätzliche Angaben, die in Unterpunkt f) der entsprechenden Funktionalitätsklasse spezifiziert sind

C.2 Abgrenzung des Untersuchungsgegenstands DRNG:

Untersuchungsgegenstand ist das beschreibende 5-Tupel (S, R, ϕ, ψ, p_A) . Die Beurteilung der Seedgenerierung, d.h. der praktischen Realisierung der Verteilung p_A (Angabe des Antragstellers) ist nicht Gegenstand der eigentlichen DRNG-Evaluation und wird von den im folgenden beschriebenen Evaluationskriterien nicht abgedeckt. Dennoch hat der Antragsteller die Generierung des Seed nachvollziehbar zu beschreiben (C.1(iv)) und zu begründen, weshalb auf diese Weise die Verteilung p_A induziert wird (siehe auch Beispiel E.7).

C.3 Allgemeine Anmerkung zur Spezifikation der Funktionalitätsklassen:

- (i) Die in Unterpunkt d) angegebene Mechanismenstärke bezieht sich ausschließlich auf logische Angriffe gegen das beschreibende 5-Tupel (S, R, ϕ, ψ, p_A) . Die Mechanismenstärke der Gesamtevaluation hängt selbstverständlich ganz erheblich von der technischen Realisierung des DRNG und dessen Einbindung in den (Gesamt-)Evaluationsgegenstand ab, da die Gesamtevaluation auch direkte Angriffe gegen die kryptographischen Algorithmen und Protokolle, die Software oder das Betriebssystem sowie hardwareorientierte Angriffe berücksichtigen muss.
- (ii) Unterpunkt d) beschreibt die klassenspezifischen Eigenschaften. Die neben C.1 (i) —(iv) für eine Evaluation erforderlichen Angaben sind in Unterpunkt f) zusammengestellt. Die übrigen Unterpunkte erhellen und begründen Auswahl und Zielsetzung dieser Anforderungen. Die Unterpunkte i) und j) (siehe Kapitel D) beschreiben und erläutern die Aufgaben des Evaluators.
- (iii) Die folgende Tabelle zeigt den Zusammenhang zwischen den Funktionalitätsklassen, Evaluationsstufen und Mechanismenstärken im Überblick.

Klasse	Beispiel (ggf. abhängig von der Wahl geeigneter Parameter)	Mindest-E-Stufe / Mechanismenstärke
K1	E.1: gewöhnlicher Zähler E.2: linearer Kongruenzgenerator E.3: lineares Schieberegister E.4: rekursiver Aufruf eines symmetrischen Verschlüsselungsverfahrens E.5: gewöhnlicher Zähler mit Hashfunktion E.6: RSA-Generator	E2 / niedrig, mittel E3 / hoch
K2	E.2-E.6	E2 / niedrig, mittel E3 / hoch
K3	E.4-E.6	E3 / mittel, hoch
K4	E.6	E3 / mittel, hoch

Klasse K1

a) qualitativ-intuitive Beschreibung der K1-spezifischen Anforderungen:

Eine aus Zufallszahlen r_1, r_2, \dots gebildete Folge von Zufallsvektoren (r_1, \dots, r_c) , (r_{c+1}, \dots, r_{2c}) , \dots , (r_{M-c+1}, \dots, r_M) soll mit hoher Wahrscheinlichkeit nur paarweise verschiedene Elemente enthalten. Statistische Eigenschaften der erzeugten Zufallsvektoren sind ohne Belang. (Die Wahl der Parameter c und ϵ (und letztlich auch M) hängen von der avisierten Anwendung ab.)

b) denkbare Anwendungen:

--- Aus den Zufallszahlen werden nichtkonstante Anteile für einen Challenge-Response-Algorithmus (z.B. im Rahmen einer Chipkarten-Terminal-Authentikation) generiert.

c) Zielsetzung:

Schutz gegen Replay-Attacken

d) Anforderungen an K1-DRNGs:

(i) Mit einer Wahrscheinlichkeit von mindestens $1-\epsilon$ sollen Zufallszahlenvektoren (r_1, \dots, r_c) , (r_{c+1}, \dots, r_{2c}) , \dots , (r_{M-c+1}, \dots, r_M) paarweise verschieden sein.

Für $\epsilon = 0$ ist die Mechanismenstärke hoch erfüllt. Ansonsten gilt

$M^2/c^2\epsilon > 2^{52}$ und $\epsilon < 2^{-16}$: Mechanismenstärke hoch;

$M^2/c^2\epsilon > 2^{32}$ und $\epsilon < 2^{-12}$: Mechanismenstärke mittel;

$M^2/c^2\epsilon > 2^{20}$: Mechanismenstärke niedrig.

e) Begründung:

Der Raum R der erzeugbaren Zufallszahlen ist nicht bei allen DRNGs identisch. So liefern die Beispiele E.1 bis E.6 in Abschnitt D $\lceil \log_2(N) \rceil$, f, 1, (üblicherweise) 64, m bzw. 1 Bit breite Zufallszahlen. Es kann daher keine universellen Kriterien für die Zufallszahlenfolgen r_1, r_2, \dots selbst geben, sondern es müssen vielmehr Zufallszahlenvektoren $(r_1, \dots, r_c), (r_{c+1}, \dots, r_{2c}), \dots$ untersucht werden. Die Zahlenwerte c , M und ϵ werden vom Antragsteller angegeben. Sie ergeben sich aus der bzw. den avisierten Anwendungen und der Bitbreite der vom DRNG erzeugten Zufallszahlen.

Lässt man die Schwierigkeiten außer acht, die eine konkrete technische Umsetzung aufwerfen würde, hängt der logische Aufwand einer Replay-Attacke gegen das beschreibende 5-Tupel (S, R, ϕ, ψ, p_A) streng monoton fallend von ϵ und, bei festgehaltenem c , streng monoton steigend von M ab. Konkret: Für eine erfolgreiche Replay-Attacke muss ein Angreifer in dem für ihn günstigsten Szenario (i.e. bei scharfer Schranke $1-\epsilon$) durchschnittlich $1/\epsilon$ viele, von verschiedenen DRNGs erzeugte Zufallsvektorfolgen beobachten und die einzelnen Glieder jeder dieser Folgen intern miteinander vergleichen.

f) vom Antragsteller neben C.1(i)–(iv) anzugeben:

(v) $c \in \mathbb{N}$ und $\epsilon \in [0, 1)$ (Eine Bestätigung für mehrere Parameterpaare (c, ϵ) ist möglich.)

(vi) mathematische Beweisführung (ggf. unter plausiblen Annahmen an ein mathematisches Modell; siehe die auch die Beispiele E.1–E.6), dass die Anforderung d)(i) erfüllt ist (Die mathematische Beweisführung ist optional, falls $M |A| < 2^{32}$ mit $A := \{s \in S \mid p_A(s) > 0\}$ oder $10M/\epsilon < 2^{32}$; siehe auch K1.i) (ii.b) und (ii.c) in Kapitel D.).

g) Erläuterungen: ---

h) K1-DRNGs (Beispiele):

E.1, E.2, E.3, E.4, E.5, E.6.

Klasse K2

a) qualitativ-intuitive Beschreibung der K2-spezifischen Anforderungen:

Die erzeugten Zufallszahlen besitzen ähnliche statistische Eigenschaften wie Zufallszahlen, die von einem idealen Zufallszahlengenerator erzeugt werden.

b) denkbare Anwendungen:

--- Stromchiffren, die durch ein Schieberegisterbündel gesteuert werden, deren Anfangsbelegung sich aus geheimgehaltenen Langzeitschlüsseln und einem Spruchschlüssel ergibt, welcher zu Beginn des Kommunikation offen übertragen wird.

c) Zielsetzung:

Es sollen Korrelationsattacken gegen kryptographische Algorithmen ausgeschlossen werden, die auf statistischen Schwächen der verwendeten Zufallszahlen (etwa als zufällige Schlüssel) basieren.

d) Anforderungen an K2-DRNGs:

--- Der DRNG gehört der Klasse K1 an (Abwärtskompatibilität).

(ii) Als Binärstring aufgefasst, passieren Zufallszahlenfolgen r_1, r_2, \dots und deren Projektion auf einzelne Bits die in Kapitel F spezifizierten statistischen Tests T1-T5 (siehe K2.i)).

Die Mechanismenstärke entspricht der des K1-spezifischen Anteils. Die Auswertung der Testergebnisse ist von der Mechanismenstärke unabhängig.

e) Begründung:

Wendete man die in Kapitel F spezifizierten Tests auf eine ideale Rauschquelle an, läge die Verwerfungswahrscheinlichkeit für jeden einzelnen Test bei 10^{-6} . Insbesondere läge die Wahrscheinlichkeit für eine irrtümliche Nichtanerkennung der K2-Eigenschaft (siehe K2.i)), Entscheidungsregel) unter $2.5 \cdot 10^{-6}$. Daher sollten halbwegs „vernünftige“ DRNGs die statistischen Tests praktisch immer passieren (siehe hierzu auch Bemerkung D.2). Obwohl sie nicht allzu scharf sind, dürften die statistischen Tests stark genug sein, um die bekannten Attacken gegen die kryptographischen Algorithmen auszuschließen, die auf statistischen Schwächen von Zufallszahlen basieren.

“Grundbausteine“ stochastischer Simulationen sind normalerweise sogenannte Standardzufallszahlen, die sich unter vielerlei statistischen Aspekten ähnlich verhalten wie Realisierungen unabhängiger, auf dem Intervall $[0,1)$ gleichverteilter Zufallsvariablen. Bei fast allen praxisrelevanten Problemstellungen haben nur die höchstwertigen Bits der erzeugten Standardzufallszahlen einen nennenswerten Einfluss auf das Simulationsergebnis. Auf kryptographische Anwendungen trifft dies in aller Regel nicht zu, so dass eine „Gleichartigkeit“ der einzelnen Zufallszahlenbits anzustreben ist. Die unter K2.i (iii.b) beschriebenen Tests sollen etwaige Schwächen einzelner Bits erkennen (siehe auch Beispiel E.2).

f) vom Antragsteller neben C.1(i)--(iv) und K1.f(v)-(vi) anzugeben: ---

g) Erläuterungen: ---

h) K2-DRNGs (Beispiele):

Die K2-spezifischen Anforderungen werden durch statistische Tests nachgewiesen, so dass fallspezifische theoretische Betrachtungen eigentlich entbehrlich sind. Dennoch wird in Kapitel E auch hierauf eingegangen. Man darf davon ausgehen, dass die Beispiele E.2—E.6 bei geeigneter Wahl der Parameter die erforderlichen statistischen Tests passieren.

Klasse K3

a) qualitativ-intuitive Beschreibung der K3-spezifischen Anforderungen:

Es ist einem Angreifer nicht praktisch möglich, zu einer ihm bekannten Teilfolge $r_i, r_{i+1}, \dots, r_{i+j}$ Vorgänger oder Nachfolger dieser Zufallszahlenteilfolge oder gar einen inneren Zustand zu errechnen oder zu erraten. Das dem Angreifer zugebilligte Angriffspotential hängt von der Mechanismenstärke ab.

b) denkbare Anwendungen:

- Erzeugung von Signaturschlüsselpaaren
- Erzeugung von DSS-Signaturen (privater Schlüssel x oder Zufallszahl k ; siehe [FI186])
- Erzeugung von Spruchschlüsseln für symmetrische Verschlüsselungsverfahren
- pseudozufällige Paddingbits (siehe auch [RSA], Abschnitt 8.1)
- zero-knowledge-proofs

c) Zielsetzung:

Schutz gegen Rekonstruierbarkeit alter und Vorhersagbarkeit zukünftiger Zufallszahlen bei Kenntnis einer Teilfolge.

d) Anforderungen an K3- DRNGs:

- Der DRNG gehört der Klasse K2 an (Abwärtskompatibilität).
- (iii) Mechanismenstärke hoch: $H(p_A) \geq 80$; Mechanismenstärke mittel: $H(p_A) \geq 48$. (Es bezeichnet $H(p_A) = -\sum_{s \in S} p_A(s) \log_2(p_A(s))$ die Entropie von p_A .)
- (iv) Es darf einem Angreifer nicht praktisch möglich sein, zu einer ihm bekannten Teilfolge $r_i, r_{i+1}, \dots, r_{i+j}$ den Vorgänger r_{i-1} oder den Nachfolger r_{i+j+1} zu errechnen ($i+j \leq M$), wobei das ihm zugebilligte Angriffspotential von der Mechanismenstärke abhängt. Selbst bei maximalem derzeitigem Know-how darf die Ratechance (realisiert durch eine gezielte Teilexhaustion) bestenfalls vernachlässigbar höher sein, als dies ohne Kenntnis der Teilfolge der Fall wäre. Es wird angenommen, dass der Angreifer das beschreibende 5-Tupel kennt. Er kennt jedoch keinen der inneren Zustände s_0, s_1, \dots, s_M .

Bei Mechanismenstärke hoch wird dem Angreifer das maximale derzeit öffentlich verfügbare Know-how, die Verfügbarkeit über die derzeit leistungsfähigste Technologie

in beliebigem Umfang und ein Angriffszeitraum von etlichen Jahren zugestanden. Bei Mechanismenstärke „mittel“ wird dem Angreifer mittleres Angriffspotential im Sinne von ITSEM, Anhang 6.C, zugestanden (siehe auch g) und Beispiel E.4). Eine Evaluation der K3-spezifischen Eigenschaften mit der Mechanismenstärke niedrig ist nicht möglich.

e) Begründung:

In der Definition des zu berücksichtigenden Angriffspotentials geht d)(iv) bei Mechanismenstärke hoch deutlich über die entsprechende Definition in ITSEM, Anhang 6.C, hinaus. Die in d)(iv) geforderten Einwegeigenschaften rücken K3-DRNGs in die Nähe kryptographischer Mechanismen (z.B. Hashfunktionen). Für diese ist in ITSEC 3.23 bzw. ITSEM 6.C.34 ein besonderes Bewertungsverfahren implizit vorgesehen.

Eine Erhöhung des zu berücksichtigenden Angriffspotentials über die Anforderungen von ITSEM hinaus ist insbesondere im Zusammenhang mit digitalen Signaturen und bei der Generierung symmetrischer Spruchschlüssel für sicherheitskritische Anwendungen unverzichtbar. Eine Ausnahme stellt die Erzeugung von Spruchschlüsseln für symmetrische Verschlüsselungsverfahren dar, die selbst nur mittlere Mechanismenstärke besitzen.

Die Längenbeschränkung der als bekannt vorausgesetzten Zufallszahlenteilfolge r_i, \dots, r_{i+j} ergibt sich in natürlicher Weise aus C.1 (iii). Anforderung d)(iv) garantiert gleichzeitig die Sicherheit jedes Vorgängers und jedes Nachfolgers r_v (mit $v \leq M$) dieser Teilfolge, da außer $i+j \leq M$ nichts vorausgesetzt wird. (Verlängert man die als bekannt vorausgesetzte Teilfolge zu r_{v+1}, \dots, r_{i+j} bzw. r_i, \dots, r_{v-1} , ist r_v ihr direkter Vorgänger bzw. Nachfolger). Im übrigen sichert d)(iv) auch die inneren Zustände, da bei Kenntnis von s_t die Zufallszahlen r_t, r_{t+1}, \dots mühelos zu errechnen wären. Wegen d)(iii) und der K2-Eigenschaft sollte es ohne Kenntnis einer Zufallszahlenteilfolge praktisch unmöglich sein, den inneren Zustand oder einzelne Zufallszahlen bzw. kurze Teilfolgen zu erraten, da, von pathologischen Ausnahmen abgesehen, bereits relativ kurze Zufallszahlenteilfolgen nahezu die gesamte Entropie des seed besitzen. („Kurz“ hängt von der Bitbreite der Zufallszahlen ab.)

Eine Formalisierung der K3-Anforderungen, insbesondere des Aspekts des Erratens, verlangt einen Spagat zwischen mathematischer Exaktheit und praktischer Verifizierbarkeit der Kriterien. Zahllose Arbeiten über „Hardcorebits“ bedienen sich komplexitätstheoretischer Charakterisierungen (siehe z.B. den Übersichtsartikel [La]), um Begriffe wie die Unvorhersagbarkeit von Bits (siehe auch [ACGS], 196) zu definieren. Ihr entscheidende Nachteil für den von uns avisierten Verwendungszweck besteht darin, dass Definitionen und Schlussfolgerungen sich nicht auf einen einzelnen DRNG, sondern auf eine ganze Familie von DRNGs beziehen, d.h. sie sind asymptotischer Natur. Eine quantitative Bestimmung des Rechenaufwands bzw. einer Ratechance unter Berücksichtigung eines maximal zur Verfügung stehenden Angriffspotentials dürfte im konkreten Einzelfall äußerst schwierig bzw. unmöglich sein. Stattdessen wird der pragmatische Ansatz propagiert (vgl. i)), die Verifikation der

Eigenschaft d)(iv) auf ein verwandtes Problem abzuwälzen, welches nach allgemeiner Auffassung mit dem zugebilligten Angriffspotential als nicht praktisch durchführbar gilt (wenngleich dies normalerweise nicht formal bewiesen werden kann).

f) vom Antragsteller neben C.1(i)--(iv) und K1.f(v)-(vi) anzugeben:

(vii) mathematische Beweisführung (ggf. unter plausiblen Annahmen an ein mathematisches Modell), dass Anforderung d)(iv) erfüllt ist

g) Erläuterungen

zu d)(iv), f)(vii): Der Nachweis der d)(iv)-Eigenschaft kann darin bestehen, dass, gegebenenfalls unter plausiblen Annahmen, gezeigt wird, dass ein Errechnen von r_{i-1} bzw. r_{i+j+1} oder das Angeben einer Ratestrategie, die die Kenntnis von r_i, \dots, r_{i+j} ausnutzt, mindestens ebenso schwierig ist wie ein Problem, das nach allgemeiner Auffassung mit dem in d)(iv) spezifizierten Angriffspotential als nicht praktisch durchführbar angesehen wird (siehe Beispiele E.4 und E.6).

h) K3-DRNGs (Beispiele):

Mechanismenstärke hoch: E.4 (für Enc = Triple-DES, IDEA), E.5, E.6;

Mechanismenstärke mittel: E.4 (für Enc = DES).

Klasse K4

a) qualitativ-intuitive Beschreibung der K4-spezifischen Anforderungen:

Es ist einem Angreifer nicht praktisch möglich, aus Kenntnis eines inneren Zustands s_i Vorgängerzufallszahlen oder Vorgängerzustände zu errechnen oder zu erraten. Das dem Angreifer zugebilligte Angriffspotential hängt von der Mechanismenstärke ab.

b) denkbare Anwendungen:

--- Erzeugung von Signaturschlüsselpaaren

--- Erzeugung von DSS-Signaturen (privater Schlüssel x oder Zufallszahl k ; siehe [FI186])

--- Erzeugung von Spruchschlüsseln für symmetrische Verschlüsselungsverfahren

--- pseudozufällige Paddingbits (siehe auch [RSA], Abschnitt 8.1)

c) Zielsetzung:

Schutz gegen Rekonstruierbarkeit alter Zufallszahlen bei Kenntnis eines inneren Zustands. (Hintergrund: Angreifer bringt sich in Besitz der technischen Realisierung des DRNG und ist in der Lage, den inneren Zustand auslesen.)

d) Anforderungen an K4-DRNGs:

--- Der DRNG gehört der Klasse K3 an (Abwärtskompatibilität).

(v) Es darf einem Angreifer nicht praktisch möglich sein, bei Kenntnis des inneren Zustands s_i die Vorgängerzufallszahl r_{i-1} zu errechnen, wobei das ihm zugebilligte Angriffspotential von der Mechanismenstärke abhängt. Selbst bei maximalem derzeitigem Know-how darf die Ratechance (realisiert durch eine gezielte Teilexhaustion) bestenfalls vernachlässigbar höher sein, als dies ohne Kenntnis von s_i der Fall wäre. Es wird angenommen, dass der Angreifer das beschreibende 5-Tupel kennt.

Bei Mechanismenstärke hoch wird dem Angreifer das maximale derzeit öffentlich verfügbare Know-how, die Verfügbarkeit über die derzeit leistungsfähigste Technologie in beliebigem Umfang und ein Angriffszeitraum von etlichen Jahren zugestanden. Bei Mechanismenstärke „mittel“ wird dem Angreifer mittleres Angriffspotential im Sinne von ITSEM, Anhang 6.C, zugestanden. Eine Evaluation der K4-spezifischen Eigenschaft mit der Mechanismenstärke niedrig ist nicht möglich.

e) Begründung: analog zu K3.e). Es sei lediglich bemerkt, dass die Anforderung d)(v) nicht nur den direkten Zufallszahlenvorgänger r_{i-1} , sondern jedes r_v und s_v mit $v < i$ schützt (Begründung analog zu K3.e), 3. Absatz). Anforderung d(v) verschärft die „Rückwärtseigenschaft“ aus d(iv), da man aus dem inneren Zustand s_i die Zufallszahlen r_i, \dots, r_{i+j} leicht berechnen kann.

f) vom Antragsteller neben C.1(i)--(iv), K1.f)(v)-(vi) und K3.f)(vii) anzugeben:

(vi) mathematische Beweisführung (ggf. unter plausiblen Annahmen an ein mathematisches Modell), dass die Anforderung d)(v) erfüllt ist

g) Erläuterungen:

siehe K3.g)

h) K4-DRNGs (Beispiele):

E.6 (Mechanismenstärke hoch)

C.4 Bemerkung: Es ist klar, dass man sich mit K3- oder K4-DRNGs mit hoher Mechanismenstärke stets „auf der sicheren Seite“ befindet. Andererseits genügt offenkundig bereits die Mechanismenstärke mittel, falls der DRNG etwa zur Schlüsselerzeugung für einen Verschlüsselungsalgorithmus verwendet wird, der selbst nur mittlere Mechanismenstärke genügt. Insofern die avisierten Anwendungen dies erlauben, kann es durchaus sinnvoll sein, einen K1- oder K2-Generator zu benutzen, da dieser normalerweise weniger Rechenzeit und seine Implementierung weniger Code und damit weniger RAM erfordert als dies für K3- oder K4-Generatoren der Fall ist. Vor allem bei Chipkartenanwendungen können diese Aspekte von Bedeutung sein. Hinzu kommt, dass bei der Implementierung eines K3- bzw. K4-DRNGs konsequenterweise durch geeignete Schutzmaßnahmen (Hardware, Software, Betriebssystem) dafür gesorgt werden muss (Gesamtevaluation!), dass der innere Zustand des DRNGs (z.B. ein

geheimer kryptographischer Schlüssel) zuverlässig gegen unbefugtes Auslesen geschützt ist. Für K1- und K2-DRNGs ist dies entbehrlich.

D. Evaluationsmethodologie

Kapitel D beschreibt, wie der Evaluator die spezifischen Eigenschaften der jeweiligen Funktionalitätsklasse prüfen soll. Die Nummerierung der Unterpunkte beginnt mit i)

D.0 Zusammenhang zur Gesamtevaluation: In den Sicherheitsvorgaben spezifiziert der Hersteller die Anforderungen an die Sicherheitsfunktionalität. Wenn im Einzelfall auf dieser Ebene der Abstraktion die Spezifikation der Zufallszahlenerzeugung und -nutzung bereits sinnvoll ist, wird hier die Funktionalitätsklasse für den Zufallszahlengenerator mit Bezug zur Sicherheitsfunktion des (Gesamt-)EVGs angegeben. (Oft ist der deterministische Zufallszahlengenerator nur ein Teil des zu evaluierenden Produkts.) Annahmen an die Einsatzumgebung und für die sichere Nutzung des EVGs sind zu benennen (z.B. die Forderung nach einer geeigneten seed-Generierung).

Im Feinentwurf muss die Implementierung des deterministischen Zufallszahlengenerators spezifiziert werden. Dies kann im Einzelfall Auswirkungen auf die Dokumentation zu Auslieferung und Konfiguration, Anlauf und Betrieb und die Betriebsdokumentation haben.

Insbesondere ist im Rahmen der Analyse der Eignung zu begründen, warum die Methode der seed-Generierung geeignet ist (siehe auch C1 (iv) und C.2). Die seed-Generierung sollte ebenfalls Gegenstand von Penetrationstests am EVG sein.

D.1 Umfang und Reihenfolge der Evaluationssarbeiten:

Untersuchungsgegenstand ist das beschreibende 5-Tupel (S, R, ϕ, ψ, p_A) . Die Erzeugung des seed, d.h. die praktische Realisierung des Anfangszustandes p_A , ist nicht Gegenstand der eigentlichen DRNG-Evaluation und wird von den Evaluationskriterien nicht abgedeckt (siehe C.2). Dennoch wird die eigentliche Evaluation nur dann durchgeführt, falls die Argumentation des Antragsstellers, weshalb auf diese Weise die Verteilung p_A induziert wird, den Evaluator überzeugt. Für die Evaluation selbst spielt die praktische Realisierung dann keine Rolle mehr.

- Der Evaluator prüft die vom Antragsteller in der Analyse der Eignung bezüglich der Realisierung der Anfangsverteilung p_A abgelieferte Argumentation des Antragstellers.
- Der Evaluator hat das beschreibende 5-Tupel (S, R, ϕ, ψ, p_A) mit der informellen Beschreibung des deterministischen Rauschgenerators (C.1(ii.a)) zu vergleichen und auf Konsistenz zu prüfen.
- Der Evaluator erledigt klassenspezifische Aufgaben, die Unterpunkt i) der entsprechenden Funktionalitätsklasse spezifiziert und in j) erläutert sind.

Klasse K1 (Fortsetzung)**i) Aufgaben des Evaluators:**

(ii.a) insofern vorhanden: Verifikation der vom Antragsteller gelieferten mathematischen Beweisführung $f(v)$, dass der vorgelegte DRNG zur Klasse K1 gehört.

(ii.b) falls eine mathematische Beweisführung K1-Eigenschaft fehlt und gleichzeitig $M|A| < 2^{32}$ mit $A := \{s \in S \mid p_A(s) > 0\}$ gilt, erfolgt der Nachweis der K1-Eigenschaft mittels Durchprobieren aller zulässigen Anfangswerte $s_0 \in A$, wobei jeweils die Vektoren $(r_1, \dots, r_c), \dots, (r_{M-c+1}, \dots, r_M)$ gebildet und die Folgenglieder auf paarweise Verschiedenheit geprüft werden. Die Ergebnisse werden gemäß p_A gewichtet.

(ii.c) Ist eine Verifikation der K1-Eigenschaft mittels (ii.a) oder (ii.b) nicht möglich, aber $10M/\epsilon < 2^{32}$ und die Mechanismenstärke höchstens mittel, erfolgt die Prüfung mittels einer stochastischen Simulation. Hierzu erzeugt der Evaluator gemäß p_A auf (pseudo-)zufällige Weise Anfangszustände $s_{0,1}, \dots, s_{0,t} \in S$ mit $t = \lfloor 10/\epsilon \rfloor$. Zu jedem dieser Anfangszustände berechnet er die Vektoren $(r_1, \dots, r_c), \dots, (r_{M-c+1}, \dots, r_M)$ und prüft die Folgenglieder auf paarweise Verschiedenheit.

Falls bei höchstens einer der insgesamt t Einzelsimulationen Zufallsvektoren mehrfach auftreten, wird dem DRNG die K1-Eigenschaft (mit den Parametern c, M, ϵ) bestätigt.

j) Erläuterungen zu i):

zu i)(ii.b) und (ii.c): Der Evaluator muss maximal 2^{32} Zufallszahlen erzeugen.

zu i)(ii.c) Der Evaluator fasst die Ergebnisse der t Einzelsimulationen als Realisierungen t unabhängiger, identisch $B(1, p)$ -verteilter Zufallsvariablen mit unbekanntem p auf, wobei das Ergebnis „1“ dem Mehrfachauftreten von Vektoren entspricht. Für eine Bestätigung der K1-Eigenschaft ist lediglich interessant, ob $p \leq \epsilon$ gilt. Hierzu führt der Evaluator einen statistischen Test mit Nullhypothese $H_0: p > \epsilon$ und Alternativhypothese $H_1: p \leq \epsilon$ durch, und er verwirft die Nullhypothese genau dann, falls bei weniger als zwei Einzelsimulationen Zufallsvektoren mehrfach auftreten. In Abhängigkeit von der unbekannten Wahrscheinlichkeit p beträgt die Wahrscheinlichkeit für dieses Ereignis $q(p) := (1 + \lambda_p)e^{-\lambda_p}$ mit $\lambda_p = pt$ (Poissonapproximation).

Ist $p > \epsilon$, liegt die Wahrscheinlichkeit für eine ungerechtfertigte Anerkennung der K1-Eigenschaft unterhalb von $q(\epsilon) = 0.0005$. Für $p < \epsilon/128$, beispielsweise, wird dem DRNG die K1-Eigenschaft mit einer Wahrscheinlichkeit von weniger als $q(\epsilon/128) = 0.003$ irrtümlich verweigert. (Dies bedeutet faktisch, dass der Antragsteller ein deutlich größeres ϵ als das (vermutlich) tatsächliche angeben muss, um die K1-Eigenschaft durch die Prüfvorschrift (ii.c) bescheinigt zu bekommen.)

Klasse K2 (Fortsetzung)**i) Aufgaben des Evaluators:**

--- Verifikation der K1-Eigenschaft (siehe K1.i))

Es bezeichnen f die Breite der vom DRNG erzeugbaren Zufallszahlen in Binärdarstellung (normalerweise ist $f = \log_2 \lceil |R| \rceil$) und π_w die Projektion auf die w -te Komponente.

(iii.a) Der Evaluator wählt gemäß p_A einen Anfangszustand $s_0 \in S$, erzeugt Zufallszahlen r_1, r_2, \dots , interpretiert diese als Bitstrings fester Länge. Auf die ersten 20.000 Bit dieser Folge wendet er die in Kapitel F beschriebenen Tests T1-T4 mit den angegebenen Verwerfungsgrenzen an. Ferner berechnet er die Testgrößen Z_1, \dots, Z_{5000} (siehe Test T5 in Kapitel F), bestimmt $\max_{\tau \leq 5000} \{|Z_\tau - 2500|\}$ und wählt ein τ_0 (bei mehreren Kandidaten zufällig), für das dieses Maximum angenommen wird. Anschließend wendet er auf die Teilfolge $b'_1 := b_{10001}, \dots, b'_{10000} := b_{20000}$ den Autokorrelationstest (Test T5) mit Shift τ_0 und den in Kapitel F angegebenen Verwerfungsgrenzen an.

(iii.b)(w) (Es ist $1 \leq w \leq f$.) Der Evaluator wählt gemäß p_A einen Anfangszustand $s_0 \in S$ und erzeugt Zufallszahlen $r_1, r_2, \dots, r_{20000}$. Auf die Folge der Projektionen $\pi_w(r_1), \dots, \pi_w(r_{20000})$ wendet er die in Kapitel F beschriebenen statistischen Tests T1-T4 mit den angegebenen Verwerfungsgrenzen an. Ferner berechnet er die Testgrößen Z_1, \dots, Z_{5000} (siehe Test T5 in Kapitel F), bestimmt $\max_{\tau \leq 5000} \{|Z_\tau - 2500|\}$ und wählt ein τ_0 (bei mehreren Kandidaten zufällig), für das dieses Maximum angenommen wird. Anschließend wendet er auf die Teilfolge $b'_1 := b_{10001}, \dots, b'_{10000} := b_{20000}$ den Autokorrelationstest (Test T5) mit Shift τ_0 und den in Kapitel F angegebenen Verwerfungsgrenzen an.

Entscheidungsregel:

Der Evaluator führt sukzessiv die Testvorschriften (iii.a), (iii.b)(1), (iii.b)(2), ..., (iii.b)(f), (iii.a), ... durch, bis insgesamt 257 Bitfolgen erzeugt und getestet wurden. Dem DRNG (i.e. dem 5-Tupel (S, R, ϕ, ψ, p_A)) wird bestätigt, dass er d)(ii) erfüllt, falls er alle Einzeltests passiert hat. Führte mehr als ein Einzeltest zu einer Verwerfung, wird dem DRNG die Eigenschaft d)(ii) nicht bestätigt.

Führte genau ein Einzeltest zu einer Verwerfung, ist das gesamte Testverfahren nochmals durchzuführen, und dem DRNG wird die Eigenschaft d)(ii) genau dann bestätigt, falls er beim Wiederholungsdurchgang alle Einzeltests passiert. Eine zweite Wiederholung ist nicht statthaft.

Dem 5-Tupel (S, R, ϕ, ψ, p_A) wird die Zugehörigkeit zur Klasse K2 bestätigt, falls es einen K1-DRNG beschreibt und die K2-spezifischen Tests gemäß der Entscheidungsregel aus i) passiert.

j) Erläuterungen zu i):

Die zu testenden Zufallszahlen müssen nicht mit dem Untersuchungsgegenstand selbst erzeugt werden. Der Evaluator kann hierfür ein von ihm erstelltes Simulationsprogramm benutzen. (Dies dürfte die Durchführung der Testvorschriften (iii.a) und (iii.b)

(w) im allgemeinen erheblich beschleunigen.) Die Anfangsverteilung p_A ist geeignet zu simulieren.

Klasse K3 (Fortsetzung):

i) Aufgaben des Evaluators:

--- Verifikation der K2-Eigenschaft (siehe K1.i), K2.i))

(iv) Verifikation von d)(iii)

Verifikation der vom Antragsteller erbrachten mathematischen Beweisführung, dass d)(iv) erfüllt ist.

Für die Bestätigung der K3-Zugehörigkeit mit Mechanismenstärke hoch (mittel) müssen nicht nur d)(iii) und d)(iv), sondern auch Eigenschaft d)(i) (siehe K1.d)) mit hoch (mittel) evaluiert werden.

j) Erläuterungen zu i):

zu i)(v): siehe K3.g)

Klasse K4 (Fortsetzung)

i) Aufgaben des Evaluators:

--- Verifikation der K3-Eigenschaft (siehe K1.i), K2.i), K3.i))

(vi) Verifikation der vom Antragsteller erbrachten mathematischen Beweisführung, dass d)(v) erfüllt ist.

Für die Bestätigung der K4-Zugehörigkeit mit Mechanismenstärke hoch (mittel) muss nicht nur d)(v), sondern auch die Eigenschaften d)(i) (siehe K1.d)), d)(iii) und d)(iv) (siehe K3.d)) mit mindestens hoch (mittel) evaluiert werden.

j) Erläuterungen zu i):

zu i)(vi): vgl. K3.g)

D.2 Bemerkung: Die Eigenschaften d(i) (von der Evaluationsmöglichkeit K1.h)(ii.c) abgesehen), d(iii), d(iv) und d(v) werden durch theoretische Beweisführungen verifiziert. Die Beweisführungen sind natürlich reproduzierbar, d.h. d(i), d(iii), d(iv) und d(v) sind Eigenschaften des beschreibenden 5-Tupels (S, R, ϕ, ψ, p_A) . Eine Verifizierung durch eine stochastische Simulation (vgl. K1.i)(ii.c)) oder durch statistische Tests (Eigenschaft d)(ii), vgl. K2.i)(iii.a) u. (iii.b)) ist dagegen zumindest nicht mit Sicherheit reproduzierbar, da der seed gemäß p_A zufällig ausgewählt wird. Daher ist zumindest die Eigenschaft d)(ii) keine Eigenschaft des 5-Tupels (S, R, ϕ, ψ, p_A) . Dieser prinzipiell unerfreuliche Zustand wird dadurch abgemildert, dass eine irrtümliche K1-Aner-

kennung mittels K1.h)(ii.c) sehr unwahrscheinlich ist (vgl. K1.i)) und dass eine Nichtanerkennung der d)(ii)-Eigenschaft bei „vernünftigen“ DRNGs noch weniger wahrscheinlich ist (vgl. K2.e)). Somit sind auch die empirisch erzielten Teilergebnisse der DRNG-Evaluation sozusagen „quasireproduzierbar“, was für die Zuverlässigkeit und Vertrauenswürdigkeit des Evaluationsverfahrens natürlich unerlässlich ist.

D.3 Evaluationsstufen: Die Klassen K1 und K2 sind bis zur Mechanismenstärke „mittel“ ab E2 unter zusätzlichen Angaben des Antragstellers evaluierbar. Ansonsten ist mindestens die Prüftiefe E3 erforderlich, wobei auch hier zusätzliche Angaben des Antragstellers notwendig sind.

E. Beispiele

In den Beispielen E.1-E.6 werden mehrere DRNG-Typen im Hinblick auf die in den Funktionalitätsklassen K1-K4 spezifizierten Anforderungen untersucht. Wenngleich die K2-spezifische Eigenschaft d)(ii) durch statistische Tests nachgewiesen wird, wird im folgenden auch hierauf kurz eingegangen.

E.7 liefert ein Beispiel für die geforderte Begründung C.1(iv), weshalb die Seedgenerierung die Verteilung p_A induziert.

E.0 Notation: Es bezeichnen $Z_N := \{0, 1, \dots, N-1\}$ und μ_T die Gleichverteilung auf der endlichen Menge T .

E.1 Beispiel (gewöhnlicher Zähler): Der DRNG werde durch das 5-Tupel $(Z_N, Z_N, \phi, \psi, \mu_{\{0\}})$ beschrieben mit Zustandsfunktion $\phi(j) := j+1 \pmod{N}$ und Ausgabe-funktion $\psi(j) := j$.

Ist $N > M$, so ist die K1-Eigenschaft für jedes c sogar für $\varepsilon = 0$ erfüllt. Offensichtlich scheitert der gewöhnliche Zähler selbst bei $p_A = \mu_{Z_N}$ bereits an den K2-spezifischen Tests.

E.2 Beispiel (linearer Kongruenzgenerator):

Es seien $N := 2^d$ und $a < N$ mit $a \equiv 1 \pmod{4}$. Ausgehend von einem Anfangszustand $s_0 \in Z_N$ wird eine Folge s_1, s_2, \dots rekursiv via $s_j = \phi(s_{j-1}) := (as_{j-1} + 1) \pmod{N}$ berechnet. In Schritt j werden als Zufallszahl r_j die f höchstwertigsten Bits ($f \leq d$) ausgegeben; d.h. $r_j = \psi(s_j) := \lfloor s_j / 2^{d-f} \rfloor$. Daraus ergibt sich das beschreibende 5-Tupel $(Z_N, \{0, 1\}^f, \phi, \psi, \mu_{Z_N})$.

Für $f = d$ bedeutete die Gleichheit zweier Zufallszahlenvektoren $(r_{ic+1}, \dots, r_{(i+1)c})$ und $(r_{jc+1}, \dots, r_{(j+1)c})$ insbesondere $s_{ic+1} = r_{ic+1} = r_{jc+1} = s_{jc+1}$, was für $M \leq 2^d$ (Periodenlänge des DRNG) nicht auftreten kann.

Für hinreichend großes d (z.B. $d \geq 48$) verhalten sich die Standardzufallszahlen $s_1/2^d, s_2/2^d, \dots$ in statistischer Hinsicht ähnlich wie Realisierungen unabhängiger, auf dem Intervall $[0, 1)$ gleichverteilter Zufallsvariablen. Fasst man für kleines f (und M klein

gegenüber der Periodenlänge 2^d) Zufallszahlenfolgen r_1, r_2, \dots, r_M als Realisierungen unabhängiger, auf $\{0,1\}^f$ gleichverteilter Zufallsvariablen auf, so gilt für $M/c < 2^{cf/2}$ in guter Näherung $P((r_1, \dots, r_c), \dots, (r_{M-c+1}, \dots, r_M) \text{ paarweise verschieden}) \approx e^{-M/c * (M/c-1)/2^{cf+1}}$ (Geburtstagsphänomen!). Für „vernünftiges“ M und ε erfüllt der lineare Kongruenz-generator also auch bei kleinem f die Eigenschaft d)(i). (Für $M/c = 2^{16}$ und $cf = 54$, beispielsweise, beträgt der rechte Term $\approx 1-2^{-23}$.) Für $f \approx d$ wird der lineare Kongruenz-generator die K2-spezifischen Tests nicht bestehen, da das k -niederwertigste Bit 2^k -periodisch ist.

E.3 Beispiel (lineares Schieberegister):

Es bezeichne $p: \{0,1\}^d \rightarrow \{0,1\}$, $p(x) := \sum_{j=0}^{d-1} a_j x^j$ das primitive Rückkopplungspolynom eines linearen Schieberegisters der Länge d . Der Anfangszustand, d.h. die Anfangsbelegung des Schieberegisters werde zufällig (gleichverteilt) aus der Menge der von 0 verschiedenen d -Tupel gewählt. Das beschreibende 5-Tupel ist durch $(\{0,1\}^d, \{0,1\}, \varphi, \psi, \mu_{S \setminus \{0\}})$ gegeben. Dabei sind $\varphi(b_{n-1}, \dots, b_{n+d-2}) := (b_n, \dots, b_{n+d-2}, b_{n+d-1} := \sum_{j=0}^{d-1} a_j b_{n+d-2-j})$ und $\psi(b_n, \dots, b_{n+d-1}) := b_n$.

Da p nach Voraussetzung primitiv ist, existiert neben dem Nullzustand nur ein einziger Zykel der Länge $2^d - 1$, nämlich genau die Menge aller möglichen Anfangszustände. Ist $c \geq d$ und $M \leq 2^d - 1$, so sind die (M/c) vielen c -Tupel $(r_1, \dots, r_c), (r_{c+1}, \dots, r_{2c}), \dots, (r_{M-c+1}, \dots, r_M)$ für jedes $s_0 \in S \setminus \{0\}$ paarweise verschieden, da die Gleichheit zweier c -Tupel insbesondere die Gleichheit ihrer Projektionen auf die ersten d Komponenten, d.h. die Gleichheit der inneren Zustände impliziert. Für $c \geq d$ gehört auch das lineare Schieberegister zur Klasse K1, und zwar mit dem Optimalwert $\varepsilon = 0$. Ist d hinreichend groß, sollte es auch die K2-spezifischen statistischen Tests bestehen.

Bei bekanntem Rückkopplungspolynom p benötigt man nur etwa d Zufallszahlen, um den inneren Zustand des Schieberegisters zu rekonstruieren. (Ist p unbekannt, kann es bei Kenntnis einer ungefähr $2d$ Bit umfassenden Zufallszahlenteilfolge mit dem Berlekamp-Massey-Algorithmus bestimmt werden.) Lineare Schieberegister gehören folglich nicht zur Klasse K3.

E.4 Beispiel (rekursiver Aufruf eines symmetrischen Verschlüsselungsverfahrens):

Es bezeichne Enc einen symmetrischen Blockverschlüsselungsalgorithmus (z.B. DES, Triple-DES, IDEA) mit identischem Klar- und Geheimentextraum, während S_B und S_K den Raum der Klartextblöcke bzw. den Schlüsselraum bezeichnen. Der Anfangszustand $s_0 := (r_0, k) \in S_B \times S_K$ werde zufällig (gleichverteilt) gewählt. Der Schlüssel k bleibt während der gesamten Zufallszahlenerzeugung konstant und wird geheimgehalten.

Der DRNG wird durch das 5-Tupel $(S_B \times S_K, S_B, \varphi, \psi, \mu_{S_B \times S_K})$ beschrieben mit $\varphi: S_B \times S_K \rightarrow S_B \times S_K$, $s_n = (r_n, k) = \varphi(r_{n-1}, k) := (\text{Enc}(r_{n-1}, k), k)$ und $\psi: S_B \times S_K \rightarrow S_B$, $\psi(r_n, k) := r_n$.

Die Zufallszahlen r_1, \dots, r_M sind genau dann disjunkt, falls der Anfangswert $r_0 \in S_B$ sich bezüglich der Permutation $r \rightarrow \text{Enc}(r; k)$ in einem Zykel der Länge $> M$ befindet. Für $\text{Enc} = \text{DES}$, $\text{Enc} = \text{Triple-DES}$ oder $\text{Enc} = \text{IDEA}$ darf man aufgrund des Forschungsstandes annehmen, dass die Zufallsvariable $k \rightarrow \text{Enc}(\bullet; k)$ (zufällige Schlüsselwahl!) ähnliche Eigenschaften aufweist wie eine zufällige Permutation. Unter dieser Modellannahme ist die Zyklenlänge von r_0 (zufällige Auswahl!), wie man sich leicht überlegt, gleichverteilt auf der Menge $\{1, \dots, |S_B|\}$. Also ist $P((r_1, \dots, r_c), \dots, (r_{M-c+1}, \dots, r_M) \text{ paarweise disjunkt}) \geq P(r_1, \dots, r_M \text{ paarweise disjunkt}) \approx 1 - M/|S_B|$, so dass der DRNG für $\epsilon \geq M/|S_B|$ zur Klasse K1 zu zählen ist. Da bei den allgemein als stark angesehenen Verschlüsselungsverfahren keine statistischen Auffälligkeiten bekannt sind, darf man davon ausgehen, dass dieser DRNG auch die K2-spezifischen Tests passiert.

Könnte man aus Kenntnis einer Teilfolge r_i, \dots, r_{i+j} deren Vorgänger r_{i-1} oder gar den inneren Zustand, insbesondere also den geheimen Schlüssel k bestimmen, stellte dies eine erfolgreiche (spezielle) Known-Plaintext-Attacke gegen das Verschlüsselungsverfahren Enc gegen Klartext bzw. Schlüssel dar. (Bemerkung: Eine erfolgreiche Durchführung einer solchen Attacke wäre mindestens ebenso schwierig wie die einer Chosen-Plaintext-Attacke gegen Enc .) Analog dazu stellt die Aufgabe, den Nachfolger r_{i+j+1} zu bestimmen, eine spezielle Known-Plaintext-Attacke gegen die Entschlüsselungsfunktion Enc^{-1} dar. Für $\text{Enc} = \text{DES}$, Triple-DES oder IDEA erfüllt der DRNG die K3-spezifischen Eigenschaften d)(iii) und d)(iv), da auch keine Ratestrategien bekannt sind, um unbekannte Klartext- bzw. Schlüsselbits mit einer über 0.5 liegenden Wahrscheinlichkeit zu erraten. $\text{Enc} = \text{Triple-DES}$ oder IDEA besitzen hohe, der einfache DES nur noch mittlere Mechanismenstärke (Schlüssellexhaustion!). Dieser DRNG-Typ gehört nicht zur Klasse K4 (Entschlüsselung!).

E.5 Beispiel (gewöhnlicher Zähler mit Hashfunktion):

Es seien $S = Z_N$ mit $N \geq 2^{200}$, $\varphi: Z_N \rightarrow Z_N$, $\varphi(j) := j+1 \pmod{N}$, und der Anfangswert $s_0 \in S$ werde zufällig, d.h. gleichverteilt auf Z_N gewählt und geheimgehalten.

Ferner bezeichne $H: \{0,1\}^N \rightarrow \{0,1\}^m$ eine als geeignet angesehene Hashfunktion (z.B. RIPEMD-160). Dann definiert $(Z_N, \{0,1\}^m, \varphi, H, \mu_{Z_N})$ einen DRNG.

Fasst man die Hashfunktion als Zufallsvariable über Z_N mit Werten in $\{0,1\}^m$ auf, und nimmt man ferner an, dass Folgen $H(i), H(i+1), \dots$ ähnliche statistische Eigenschaften besitzen wie Realisierungen unabhängiger, auf $\{0,1\}^m$ gleichverteilter Zufallsvariablen, so verifiziert man die K1-Eigenschaft wie in E.2 für kleines f . (Es ist $P(r_1, r_2, \dots, r_M \text{ sind paarweise verschieden}) \approx e^{-M(M-1)/2|H(S)|}$.) Der DRNG erfüllt die K3-spezifischen Eigenschaften (Einwegeigenschaft der Hashfunktion) mit hoher Mechanismenstärke, wenn gleich er offensichtlich kein K4-DRNG ist.

E.6 Beispiel (RSA-Generator): (siehe auch [La], 131)

Es seien $p \neq q$ Primzahlen, $N := pq$ und $e \in \{1, \dots, \phi(N)\}$ mit $\text{ggT}(e, \phi(N)) = 1$, wobei ϕ die Eulerfunktion bezeichnet. Die Primfaktoren p und q werden geeignet (z.B. gemäß den Empfehlungen des Maßnahmenkatalogs zum Signaturgesetz (SigG)) gewählt, ge-

heimgehalten und nach Berechnung von N (bekannt) und der Auswahl von e (nicht bekannt) vernichtet. Der Anfangszustand $s_0 = (t_0, e) \in Z_N \times B := Z_N \times \{0 < y < \phi(N) \mid \text{ggT}(y, \phi(N)) = 1\}$ werde zufällig (gleichverteilt) gewählt.

Der RSA-Generator wird durch das 5-Tupel $(Z_N \times B, \{0, 1\}, \phi, \psi, \mu_{Z_N \times B})$ beschrieben, wobei die Abbildungen ϕ und ψ durch $\phi: Z_N \times B \rightarrow Z_N \times B$, $\phi(t_{n-1}, e) := (t_{n-1}^e \pmod{N}, e)$ und $\psi: Z_N \times B \rightarrow \{0, 1\}$, $\psi(t_n, e) := t_n \pmod{2}$ gegeben sind.

Zur Beurteilung von RSA-Generatoren ziehen wir in der Literatur vorhandene, asymptotische Resultate heran, wobei wir annehmen, dass die Asymptotik bei der Größenordnung des Moduls N „greift“. Es wird insbesondere angenommen, dass das Invertieren von $x \rightarrow x^e \pmod{N}$ bei Kenntnis von e , aber ohne Kenntnis von $d := e^{-1} \pmod{\phi(N)}$ schwierig, i.e. praktisch nicht durchführbar ist. (Wäre dies möglich, wäre man in der Lage, gültige Signaturen allein aus Kenntnis des öffentlichen Schlüssels zu erzeugen.) In unserer Sprechweise: s_{i-1} kann aus s_i nicht praktisch errechnet oder erraten werden.

Wäre es möglich, bei Kenntnis des inneren Zustands (t_i, e) das niederwertigste Bit von t_{i-1} , also r_{i-1} , mit einer nichtvernachlässigbar über $1/2$ liegenden Wahrscheinlichkeit (siehe z.B. [La], 132 (Theorem 7.1)) zu erraten, könnte man t_{i-1} mit einem in [ACGS] angegebenen probabilistischen Polynomialzeitalgorithmus (polynomial in $\lceil \log_2(N) \rceil$) bestimmen. Geht man davon aus, dass Polynomialzeitalgorithmen praktisch durchführbar sind (Hypothese!), ist die K4-spezifische Eigenschaft d)(v) gezeigt: Da die Bestimmung von s_{i-1} aus s_i nach obiger Annahme über die Sicherheit einer gesetzeskonformen RSA-Signatur nicht praktisch möglich ist, kann ein Raten von r_{i-1} bei Kenntnis von r_i, \dots, r_{i+j} letztlich nicht erfolversprechender als „blindes“ Raten sein, bei dem „0“ und „1“ jeweils mit Wahrscheinlichkeit $1/2$ gewählt werden. Insbesondere ist damit auch die „Rückwärtseigenschaft“ der K3-spezifischen Anforderung d)(iv) verifiziert. Nun ist aber auch $t_v \equiv t_{v+1}^d \pmod{N}$ mit $d \equiv e^{-1} \pmod{\phi(N)}$. Die obige Argumentation gilt natürlich nicht nur für den Anfangszustand $s_0 := (t_0, e)$, sondern auch für $s_0' := (t_{i+j+1}, d)$. Da weder e noch d bekannt sind, ist damit aus Symmetriegründen auch die K3-„Vorwärtseigenschaft“ gezeigt.

Aufgrund der obigen Überlegungen (siehe auch [La], 132 (Theorem 7.1) u. 126 (Theorem 4.1)) darf man annehmen, dass sich die c -Tupel $(r_1, \dots, r_c), (r_{c+1}, \dots, r_{2c}) \dots$ unter statistischen Aspekten ähnlich wie Realisierungen unabhängiger, auf $\{0, 1\}^c$ gleichverteilter Zufallsvektoren verhalten. Die K1-Eigenschaft folgt wie in Beispiel E.2 für kleines f . Ebenso darf man davon ausgehen ([La], 126 (Theorem 4.1)), dass der RSA-Generator auch die K2-spezifischen Tests passiert. Die Verwerfungswahrscheinlichkeit dürfte in der Größenordnung des Fehlers 1. Art anzusiedeln sein.

E.7 Seedgenerierung

Es sei $p_A = \mu_S$ mit $S = \{0, 1\}^{128}$. Der seed errechnet sich aus Tastatureingaben des Anwenders vor der Erstnutzung des Evaluationsgegenstands. Zulässige Eingaben sind Groß- und Kleinbuchstaben, die Ziffern 0-9 sowie „.“ und „:“, insgesamt als 64 Zei-

chen. Jedes eingegebene Zeichen wird als 6-Bit-Wort codiert und aufeinanderfolgende 6-Bit-Worte werden konkateniert. Gleicht ein Zeichen seinen beiden Vorgängern, wird es ignoriert. Der aus den ersten 85 nichtignorierten Zeichen entstandene Bitstring (510 Bit) wird mit RIPEMD-160 gehasht. Die ersten 128 Bit des Hashwerts liefern den seed. Der Anwender wird im Benutzerhandbuch informiert, dass die Zeichenkette möglichst „chaotisch“ sein soll.

Beurteilung: Das Verfahren zur Generierung des seeds ist geeignet, $p_A = \mu_S$ zu realisieren. In der Tat dürfte bereits ein Entropiezuwachs pro nichtignoriertem Zeichen in der Größenordnung von 1.5 Bit hinreichend sein.

F. Statistische Tests

Nachfolgend sind die statistischen Tests aufgelistet, die zur Verifikation der K2-spezifischen Eigenschaft d)(ii) durchgeführt werden. Die Tests T1 – T4 sind samt Bezeichnung und Verwerfungsgrenzen [FI140] (4.11.1) entnommen.

Es bezeichnet b_1, \dots, b_{20000} eine Bitfolge der Länge 20000. Wurde die Folge b_1, \dots, b_{20000} von einer idealen Rauschquelle erzeugt, liegt die Verwerfungswahrscheinlichkeit jedes einzelnen Tests bei 10^{-6} .

Test T1 (Monobittest)

$$X := \sum_{j=1}^{20000} b_j$$

Die Folge b_1, \dots, b_{20000} passiert den Monobittest, falls $9654 < X < 10346$.

Test T2 (Pokertest)

Für $j = 1, \dots, 5000$ sei $c_j = 8 \cdot b_{4j-3} + 4 \cdot b_{4j-2} + 2 \cdot b_{4j-1} + b_{4j}$. Ferner bezeichnet $f[i] := |\{j: c_j = i\}|$.

$$Y := (16/5000)(\sum_{i=0}^{15} f[i]^2) - 5000$$

Die Folge b_1, \dots, b_{20000} passiert den Pokertest ($=\chi^2$ -Anpassungstest mit 15 Freiheitsgraden), falls $1.03 < Y < 57.4$.

Test T3 (Runtest)

Ein Run bezeichnet eine maximale Teilfolge aufeinanderfolgender Nullen bzw. Einsen.

Die Folge b_1, \dots, b_{20000} passiert den Runtest, falls die Anzahl der auftretenden Runlängen innerhalb der zulässigen Intervalle liegen, die nachfolgend spezifiziert werden. Die Null- und Einsruns werden getrennt ausgewertet.

Runlänge	zulässiges Intervall
1	2267-2733
2	1079-1421
3	502-748

4	233-402
5	90-223
≥ 6	90-233

Test T4 (Long Runtest)

Ein Run der Länge ≥ 34 wird als Long Run bezeichnet.

Die Folge b_1, \dots, b_{20000} passiert den Long Runtest, falls kein Long Run auftritt.

Test T5 (Autokorrelationstest)

Für $\tau \in \{1, \dots, 5000\}$ ist $Z_\tau := \sum_{j=1}^{5000} (b_j \oplus b_{j+\tau})$

Die Folge b_1, \dots, b_{20000} passiert den Autokorrelationstest (mit Shift τ), falls $2326 < Z_\tau < 2674$. (Man beachte, dass die Teilfolge $b_{10001}, \dots, b_{20000}$ nicht in die Testgröße eingeht.)

G. Literatur

- [ACGS] W. Alexi, B. Chor, O. Goldreich and C.P. Schnorr: RSA and Rabin Functions. Certain parts are as hard as the whole. SIAM J. Comput. **17** (no. 2), April 1990, 194--209.
- [FI140] FIPS PUB 140-1 (January 11, 1994), NIST, Security Requirements for Cryptographic Modules.
- [FI186] FIPS PUB 186-1 (December 15, 1998), NIST, Specifications for the Digital Signature Standard (DSS).
- [IEP] IEEE P 1363 Standard (August 22, 1996; Working Draft), Standard Specifications for Public Key Cryptography – Annex G: Cryptographic Random Numbers.
- [La] J.C. Lagarias: Pseudorandom Number Generators in Cryptology and Number Theory. Proceedings of Symposia in Applied Mathematics **42**, 1990, 115--143.
- [RSA] PKCS#1: RSA Encryption Standard. An RSA Laboratories Technical Note, Version 1.5, November 1, 1993.