ZERTIFIZIERUNGSBERICHT

Zertifizierungs-Vorgang:	**TUVIT-TSZ-CC-9261-2008**
Produkt / System:	Java Funktionsbibliothek zur Zufallszahlengenerierung RNG.java, Version 1.21.2.3
Hersteller:	Casinoland GmbH Hans-Böckler-Allee 5 30173 Hannover
Auftraggeber:	siehe oben
Prüfstelle:	datenschutz nord GmbH, Prüfstelle für IT Sicherheit
Prüfbericht:	*Version 0.3 vom 12.01.2007* Dokument-Nummer: *Casinoland_ETR_v03_20070112.doc* Autor: Dr. Sönke Maseberg, Günther Diederich
Ergebnis:	EAL3 mit Zusatz ADV_LLD.1, ADV_IMP.1 und ALC_TAT.1
Evaluierungsauflagen:	keine
Prüfbegleiter:	Dr. Christoph Sutter
Zertifizierungsauflagen:	keine

Essen, den 31.01.2008

Joachim Faulhaber

Dr. Christoph Sutter

Inhalt

Teil A

Zertifikat und Informationen zur Zertifizierung

Teil A enthält eine Kopie des ausgestellten Zertifikats sowie Informationen

- zur Zertifizierungsstelle,
- zum Zertifizierungsvorgang, und
- zu den Projektdaten der Evaluierung und Zertifizierung.

1 Das Zertifikat (erste Seite)

Zertifikat

Die Zertifizierungsstelle der TÜV Informationstechnik GmbH bescheinigt hiermit dem Unternehmen

Casinoland GmbH

dass die Java Funktionsbibliothek zur Zufallszahlengenerierung

RNG.java, Version 1.21.2.3

durch eine akkreditierte und lizenzierte Prüfstelle nach den Common Criteria (CC), Version 2.3 unter Nutzung der Evaluationsmethodologie Common Methodology for IT Security Evaluation (CEM), Version 2.3 evaluiert wurde und die Anforderungen

ISO/IEC 15408:2005 (CC V2.3)
EAL3 mit Zusatz

erfüllt.

Die Anlage ist Bestandteil des Zertifikats und besteht aus 2 Seiten.

Dieses Zertifikat gilt nur in Verbindung mit dem vollständigen Zertifizierungsbericht zur Registriernummer für die darin aufgeführten Konfigurationen und Einsatzbedingungen.

Das Zertifikat berechtigt zur Nutzung des Prüfzeichens.

CC V2.3 TSZ-CC-9261-2008 TÜViT 2008 Trusted Product
Voluntary Validation
© 2008 TÜViT GmbH - Member of TÜV NORD Group

Zertifikat-Registrier-Nr.:
TUVIT-TSZ-CC-9261-2008

Essen, 31.01.2008 gez. Faulhaber
Zertifizierungsstelle

TÜV Informationstechnik GmbH
Unternehmensgruppe TÜV NORD
Langemarckstraße 20
45141 Essen
www.certuvit.de

Akkreditiert für Zertifizierungen im Bereich IT-Sicherheit durch die DATech in der TGA GmbH Reg-Nr. DAT-ZE-014/99-01

2 Zertifizierungsstelle – CERTÜViT

CERTÜViT, die Zertifizierungsstelle der *TÜV Informationstechnik GmbH*[1] – Unternehmensgruppe TÜV NORD – wurde 1998 gegründet und bietet ein breites Spektrum von Dienstleistungen im Bereich von Sicherheitsevaluierungen und –validierungen an.

CERTÜViT ist seit September 1999 akkreditiert für Zertifizierungen von IT-Sicherheitsprodukten gemäß ITSEC und Common Criteria durch die *Deutsche Akkreditierungsstelle für Informations- und Telekommunikationstechnik (Dekitz),* nun *Deutsche Akkreditierungsstelle Technik in der TGA GmbH (DATech), Frankfurt/Main* unter der DAR-Registrierungsnummer DAT-ZE-014/99-01 und führt ihre Projekte auf Basis eines Qualitätsmanagement-Systems durch, das auf der Grundlage der ISO 9001 zertifiziert ist.

3 Spezifikation des Zertifizierungsvorgangs

Die Zertifizierungsstelle führt die Zertifizierung nach Maßgabe der folgenden Vorgaben durch:

- DIN EN 45011
- TÜViT Zertifizierungsschema
- TÜViT Zertifizierungsbedingungen
- Gemeinsame Kriterien für die Prüfung und Bewertung der Sicherheit von Informationstechnik (CC):
  - Common Criteria for Information Technology Security Evaluation (CC) part 1-3, version 2.3, August 2005.
  - Common Methodology for Information Technology Security Evaluation (CEM), version 2.3, August 2005.
- Anwendungshinweise und Interpretationen zum Schema (AIS), herausgegeben vom Bundesamt für Sicherheit in der Informationstechnik[2].

[1] Im Folgenden TÜViT genannt.

[2] Im Folgenden BSI genannt.

4 Durchführung der Evaluierung und Zertifizierung

Die Zertifizierungsstelle führt für jede einzelne Evaluierung eine Prüfbegleitung durch, um einheitliches Vorgehen, einheitliche Interpretation der Kriterienwerke und einheitliche Bewertungen sicherzustellen. Das Produkt RNG.java, Version 1.21.2.3 hat das Zertifizierungsverfahren bei der Zertifizierungsstelle der TÜViT durchlaufen. Es handelte sich hierbei um eine Re-Zertifizierung zur Erst-Zertifizierung TUVIT-TSZ-CC-9259-2007 aufgrund einer Änderung in der Einsatzumgebung. Anstelle der Sun JDK-Version 1.4.2_08 wird die Version 1.5.0_10 eingesetzt. Ferner wurden die Sicherheitsvorgaben und die Handbücher diesbezüglich aktualisiert.

Die Erst-Evaluation des Produktes RNG.java, Version 1.21.2.3 wurde bei der Prüfstelle für IT-Sicherheit der *datenschutz nord* durchgeführt und am 12.01.2007 abgeschlossen. Die Prüfstelle der *datenschutz nord* ist eine von CERTÜViT lizenzierte Prüfstelle.

Für die Re-Zertifizierung war keine Re-Evaluation notwendig, da lediglich die Sun JDK-Version geändert worden ist und vom Hersteller die erfolgreiche Durchführung aller für diese Änderung relevanten Tests der Zertifizierungsstelle nachgewiesen worden ist.

Antragsteller und Entwickler ist *Casinoland GmbH*. Vertreiber des Produktes ist *Casinoland GmbH*.

Den Abschluss der Zertifizierung bilden

- die Vergleichbarkeitsprüfung und
- die Erstellung des vorliegenden Zertifizierungsberichts.

Diese Arbeiten wurden am 31.01.2008 abgeschlossen. Das bestätigte Vertrauenswürdigkeitspaket (EAL) gilt nur unter der Voraussetzung, dass

- alle Auflagen bzgl. Generierung, Konfiguration und Betrieb, soweit sie im Teil B des Berichts angegeben sind, beachtet werden,
- das Produkt in der beschriebenen Umgebung - sofern im nachfolgenden Bericht angegeben - betrieben wird.

Dieser Zertifizierungsbericht gilt nur für die hier angegebene Version des Produktes. Die Gültigkeit kann auf neue Versionen und Releases des Produktes ausgedehnt werden, sofern der Antragsteller eine Re-Zertifizierung des geänderten Produktes entsprechend den Vorgaben beantragt und die Prüfung keine sicherheitstechnischen Mängel ergibt.

Hinsichtlich der Bedeutung der Vertrauenswürdigkeitsstufen (EAL) und der bestätigten Stärke der Funktionen (SoF) vgl. die Auszüge aus den technischen Regelwerken im Teil C dieses Zertifizierungsberichts.

5 Veröffentlichung

Der nachfolgende Zertifizierungsbericht enthält die Seiten B-1 bis B-14. Das Zertifikat und der Zertifizierungsbericht zum Produkt RNG.java, Version 1.21.2.3 wird in der TÜViT-Zertifizierungsliste (http://www.certuvit.de) aufgenommen. Diese Liste wird regelmäßig veröffentlicht.

Weitere Exemplare des vorliegenden Zertifizierungsberichts können beim Hersteller des Produktes angefordert werden. Unter der o. g. Internetadresse von CERTÜViT kann der Zertifizierungsbericht auch in elektronischer Form abgerufen werden.

Teil B

Ergebnis der Zertifizierung

Der nachfolgende Bericht ist eine Zusammenfassung aus

- den Sicherheitsvorgaben des Antragstellers für den Evaluationsgegenstand,
- den entsprechenden Prüfergebnissen des Prüflabors und
- ergänzenden Hinweisen und Auflagen der Zertifizierungsstelle.

Inhalt des Zertifizierungsergebnisses

1 Zusammenfassung

1.1 Evaluierungsgegenstand und Hintergrund der Evaluierung

Der Evaluationsgegenstand (EVG) ist die in Java realisierte Funktionsbibliothek *RNG.java, Version 1.21.2.3* eines deterministischen Zufallszahlengenerators (DRNG) der Funktionalitätsklasse K3 der [AIS 20], der z. B. auf Online-Spielsystemen eingesetzt werden kann. Da es sich bei einem Einsatz auf solchen Systemen möglicherweise um Glücksspiele mit Geldeinsatz handeln kann, müssen an die erzeugten Zufallszahlen hohe Anforderungen an die Qualität und Sicherheit gestellt werden.

Ein deterministischer Zufallszahlengenerator erzeugt auf eine deterministische Weise eine Folge von Zahlen, die einzig und allein vom Anfangszustand (Seed) und dem des DRNG zugrunde liegenden Algorithmus abhängen. Diese Zahlen sollen für einen externen Betrachter die gleichen Eigenschaften haben wie echte Zufallszahlen. Man spricht von den erzeugten Zahlen auch von Pseudozufallszahlen, da jemand, der den Seed und den Algorithmus des DRNG kennt, in der Lage ist, dieselbe Zahlenfolge zu reproduzieren. Bei einem DRNG der Funktionalitätsklasse K3 ist gewährleistet, dass es einem externen Betrachter praktisch nicht möglich ist, zu einer ihm bekannten Teilfolge der Zufallszahlen, Vorgänger oder Nachfolger dieser Zufallszahlenteilfolge oder gar einen inneren Zustand des DRNG zu errechnen oder zu erraten.

Der EVG *RNG.java, Version 1.21.2.3* erzeugt auf einem Linux-System auf deterministische Weise Zahlen zwischen 0 und 2^{31}-2. Er besitzt zwei Hauptkomponenten, den Mersenne Twister [MT 19937] und die Ausgabefunktion (vergl. auch Abschnitt 2.1 der Sicherheitsvorgaben [ST] im Teil D dieses Zertifizierungsberichtes). Mit dem Mersenne Twister wird bei Neustart des DRNG einmalig aus dem Seed der erste innere Zustand und im weiteren Betrieb die weiteren inneren Zustände berechnet. Die Ausgabefunktion berechnet aus dem aktuellen inneren Zustand die an der externen Schnittstelle zur Verfügung gestellte Zufallszahl.

Für die Generierung des Seed des EVG werden Zufallszahlen mit genügender Entropie dem Linux-Entropiepool „`/dev/random`“ entnommen, so dass es selbst bei Kenntnis des Algorithmus nicht möglich ist, dieselbe Zahlenfolge zu erzeugen wie der EVG. Darüber hinaus ist es auf Grund des Designs des EVG nicht möglich, selbst bei Kenntnis beliebig vieler Ausgaben des EVG, eine weitere Ausgabe einer Zufallszahl mit einer Wahrscheinlichkeit signifikant größer als $1/2^{31}$ vorherzusagen.

Der Auftraggeber, Entwickler und Hersteller ist “*Casinoland GmbH*, *Hans-Böckler-Allee 5*, *30173 Hannover*“.

Die Erst-Evaluation des EVG wurde auf den Grundlagen der Sicherheitsvorgaben (vgl. Teil D dieses Berichts) von der Prüfstelle der *datenschutz nord GmbH* (*datenschutz nord*)

durchgeführt und am 12.01.2007 abgeschlossen. Für diese Re-Zertifizierung war keine Re-Evaluierung notwendig (siehe Kapitel 4 in Abschnitt A dieses Berichtes).

1.2 Vertrauenswürdigkeitspaket

Die EVG Vertrauenswürdigkeitsanforderungen basieren vollständig auf den Vertrauenswürdigkeitskomponenten und –klassen in den Common Criteria, Teil 3 (vgl. Teil C dieses Berichts bzw. [CC] Teil 3 für weitergehende Informationen). Der EVG erfüllt die Vertrauenswürdigkeitsanforderungen der Stufe EAL 3 (Vertrauenswürdigkeitsstufe 3) mit Zusatz ADV_LLD.1 (Entwicklung – Beschreibender Entwurf auf niedriger Ebene), ADV_IMP.1 (Entwicklung – Teilmenge der Implementierung der TSF) und ALC_TAT.1 (Lebenszyklus-Unterstützung – Klar festgelegte Entwicklungswerkzeuge).

1.3 Stärke der Funktionen

Die Stärke der Funktionen des EVGs wird mit “hoch” (SOF-hoch) bewertet. Informationen zu den mit SOF-hoch bewerteten Sicherheitsfunktionen sind in Kapitel 9 enthalten.

1.4 Funktionalität

Die EVG Sicherheitsanforderungen enthalten Anforderungen, die nicht aus dem Teil 2 der CC entnommen sind (die Auswahl ist CC Teil 2 erweitert) [CC]. Sie können in die folgende Kategorie eingeteilt werden:

- Kyptografische Unterstützung.

In Kapitel 9 werden die funktionalen Sicherheitsanforderungen im Detail aufgeführt. Sie werden durch eine Sicherheitsfunktion des EVG realisiert:

Sicherheitsfunktion	Beschreibung
SF1	Generierung von Zufallszahlen der Funktionalitätsklasse K3 gemäß [AIS 20] mit Mechanismenstärke „hoch“ inkl. einer transition-function[3], die eine Zufallszahl aus dem Bereich [0, 2^{31}-2] liefert.

Tabelle 1: Sicherheitsfunktionen

[3] Zur Definition der “transition-function” siehe Abschnitt 2.1 und die darin enthaltene Abbildung 1 der Sicherheitsvorgaben [ST] im Teil D dieses Zertifizierungsberichtes.

1.5 Zusammenfassung der Bedrohungen und der organisatorischen Sicherheitspolitik

Aufgrund der Annahmen (siehe Abschnitt 4.1) hat ein Angreifer keinen direkten Zugriff auf das System, auf welchem der EVG installiert ist. Als Grundlage für einen Angriff stehen einem Angreifer nur die Ausgaben des EVG zur Verfügung. Da ein erfolgreicher Angriff bei einem Einsatz des EVG z. B. auf einem Online-Spielsystem einen großen finanziellen Gewinn für einen Angreifer bedeuten würde, ist davon auszugehen, dass ein Angreifer die zur Zeit verfügbaren technischen Möglichkeiten hat und bereit ist, einen großen zeitlichen Aufwand für einen Angriff zu betreiben.

Ziel eines Angreifers ist es, die Ausgaben des RNG vorherzubestimmen, wobei frühere Ausgaben des EVG kein Geheimnis darstellen. Daraus leitet sich die folgende einzige Bedrohung des EVG ab:

Bedrohung	Beschreibung
T.1	Ein Angreifer von außerhalb ist in der Lage auf Grund gesammelter Ausgaben des EVG die nächste Ausgabe vorherzusagen.

Tabelle 2: Bedrohungen

In den Sicherheitsvorgaben sind keine organisatorischen Sicherheitspolitiken definiert. Eine detaillierte Beschreibung der Bedrohungen befindet sich in den Sicherheitsvorgaben im Teil D dieses Zertifizierungsberichts.

1.6 Spezielle Konfigurationsanforderungen

Der Evaluationsgegenstand kann nur in einer Konfiguration betrieben werden. Hinweise zu Einrichtung der Betriebsumgebung finden sich in den zum Lieferumfang gehörigen Handbüchern, welchen in Kapitel 6 angegeben sind.

1.7 Annahmen über die Einsatzumgebung

Es wird angenommen, dass der Evaluationsgegenstand auf einem Redhat Linux-System mit Kernel-Version 2.6.9* in der Distribution Red Hat Linux ES 4 x86 und unter Verwendung der Java-VM Sun JDK-Version 1.5.0_10 betrieben wird.

Weitere Annahmen zum sicheren Einsatz des EVG werden in den Sicherheitsvorgaben im Teil D und im Abschnitt 4.1 des Zertifizierungsberichts beschrieben.

1.8 Unabhängigkeit des Prüfbegleiters

Der Prüfbegleiter hat innerhalb der letzten 2 Jahre für das die Zertifizierung beauftragende Unternehmen keine Beratungen oder sonstige Dienstleistungen erbracht und mit diesem Unternehmen auch keine Beziehungen gepflegt, die seine Beurteilung beeinflussen könnten.

Der Prüfbegleiter ist zu keiner Zeit an Prüfverfahren für das dem Zertifizierungsvorgang zugrunde liegende Produkt beteiligt gewesen.

1.9 Gewährleistungsausschluss

Dieses Zertifikat ist keine generelle Empfehlung des IT-Produktes durch TÜV Informationstechnik GmbH oder eine andere Organisation, die dieses Zertifikat anerkennt oder vertritt. Eine Gewährleistung für das IT-Produkt durch TÜV Informationstechnik GmbH oder eine andere Organisation, die dieses Zertifikat anerkennt oder vertritt, wird hiermit weder gegeben noch zum Ausdruck gebracht.

2 Identifikation des EVG

Der Evaluationsgegenstand (EVG) ist die in Java realisierte Funktionsbibliothek *RNG.java, Version 1.21.2.3* eines deterministischen Zufallszahlengenerators (DRNG) der Funktionalitätsklasse K3 der [AIS 20].

Die Auslieferung des EVG erfolgt per Post oder durch persönliche Übergabe auf einer CD-ROM mit den folgenden Bestandteilen:

- Java-Funktionsbibliothek *RNG.java, Version 1.21.2.3*,
- Systemverwalterhandbuch – Casinoland-RNG (Zufallszahlengenerator RNG.java), Version 1.7, 19.12.2007, Autor: Casinoland GmbH [SD] und
- Benutzerhandbuch – Casinoland-RNG (Zufallszahlengenerator RNG.java), Version 1.7, 19.12.2007, Autor: Casinoland GmbH [BD].

Darüber hinaus wird die EVG-Datei mit einer elektronischen Signatur versehen, so dass sich der Empfänger durch Verifikation der Signatur oder Hashwertvergleich von der Integrität des EVG überzeugen kann.

3 Sicherheitspolitik

Die einzige Sicherheitspolitik, die der Evaluationsgegenstand umsetzt ist die Erzeugung von deterministischen Zufallszahlen gemäß der Funktionalitätsklasse K3 der [AIS 20].

4 Annahmen und Klärung des Einsatzbereichs

4.1 Annahmen über den Einsatz

Damit die Integrität des EVG geschützt bleibt und die Sicherheit für seinen Einsatz gewährleistet ist, müssen die folgenden Annahmen über den Einsatz des EVG erfüllt sein:

Annahme	Beschreibung
A.1	Es wird angenommen, dass der EVG so betrieben wird, dass nur befugte Personen physikalischen Zugriff haben.
A.2	Der Zugriff auf das System und den EVG wird durch ein restriktives Berechtigungskonzept geschützt, so dass nur befugte Personen logischen Zugriff haben.
A.3	Es wird angenommen, dass die Integrität des Systems bei der Installation und während des Betriebs durch organisatorische Maßnahmen sichergestellt wird.
A.4	Es wird angenommen, dass bei der Installation sichergestellt wird, dass der zertifizierte EVG zum Einsatz kommt.
A.5	Es wird angenommen, dass der EVG mit einem Seed von ausreichend hoher Entropie ($H(p_A) \geq 80$) initialisiert wird, welcher durch die Verwendung eines Redhat Linux-System mit Kernel-Version 2.6.9* in der Distribution Red Hat Linux ES 4 x86 bereitgestellt wird.
A.6	Es wird angenommen, dass die EVG-Umgebung den Hash-Algorithmus SHA-512 durch die Verwendung einer Java-VM Sun JDK-Version 1.5.0_10 bereitstellt.
A.7	Es wird angenommen, dass die Integrität des Systems inklusive der Java-Bibliothek „java.security.MessageDigest“ und des EVG regelmäßig überprüft wird.
A.8	Es wird angenommen, dass die eingesetzten Administratoren vertrauenswürdig sind.

Tabelle 3: Annahmen über den Einsatz

4.2 Angenommene Einsatzumgebung

Es wird angenommen, dass der EVG in der in Abschnitt 1.7 beschriebenen Betriebsumgebung eingesetzt wird. Siehe auch die Annahmen A.5 und A.6 aus dem vorherigen Abschnitt.

4.3 Klärung des Einsatzbereichs

Der Betrieb des Evaluationsgegenstandes sieht eine integere Plattform vor. Ferner muss sichergestellt werden, dass nur befugte Personen auf den EVG und die unterliegende Plattform zugreifen können. Siehe dazu insbesondere die Annahmen in Abschnitt 3.1 der Sicherheitsvorgaben im Teil D dieses Zertifizierungsberichts.

5 Informationen zur Architektur

Der Evaluationsgegenstand besitzt zwei Hauptkomponenten, den Mersenne Twister [MT 19937] und die Ausgabefunktion. Mit dem Mersenne Twister wird bei Neustart des DRNG einmalig aus dem Seed der erste innere Zustand und im weiteren Betrieb die weiteren inneren Zustände berechnet. Die Ausgabefunktion berechnet aus den inneren Zuständen die an der externen Schnittstelle zur Verfügung gestellten Zufallszahlen. Für mehr Details siehe Abschnitt 2.1 der Sicherheitsvorgaben im Teil D dieses Zertifizierungsberichts.

6 Dokumentation

Die nachstehend genannten Dokumente werden zusammen mit dem Produkt an den Kunden ausgeliefert:

- Systemverwalterhandbuch – Casinoland-RNG (Zufallszahlengenerator RNG.java), Version 1.7, 19.12.2007, Autor: Casinoland GmbH und
- Benutzerhandbuch – Casinoland-RNG (Zufallszahlengenerator RNG.java), Version 1.7, 19.12.2007, Autor: Casinoland GmbH,

7 Testverfahren

Der EVG wurde im Rahmen der Erst-Zertifizierung vom Hersteller in der in den Sicherheitsvorgaben [ST] spezifizierten Einsatzumgebung getestet. Bei den Herstellertests wurde geprüft, ob sich der EVG als Funktionsbibliothek an seiner externen Schnittstelle wie in den Sicherheitsvorgaben (ST), in der Funktionalen Spezifikation (FSP) und dem Entwurf auf hoher Ebene (HLD) beschrieben verhält. Alle Testergebnisse stellten sich wie erwartet ein.

Vom Evaluator wurden im Rahmen der Erst-Zertifizierung die Herstellertests teilweise wiederholt und eigene Tests durchgeführt, wobei die in der [AIS 20] für die Funktionalitätsklasse K3 mit Mechanismenstärke „hoch“ vorgeschriebenen Tests

berücksichtigt wurden. Alle Testergebnisse zeigen, dass sich der Evaluationsgegenstand wie beschrieben verhält.

Außer den Tests gemäß [AIS 20] waren keine Penetrationstests notwendig, da von den Evaluatoren die aufgeführten Gegenmaßnahmen als hinreichend angesehen wurden, um alle offensichtlichen Schwachstellen unwirksam zu machen.

Für diese Re-Zertifizierung wurden vom Hersteller die für die Änderung der Einsatzumgebung relevanten Testfälle erfolgreich wiederholt.

8 Evaluierte Konfiguration

Der EVG wird in einer festen Konfiguration ausgeliefert und in dieser Form installiert. Daher ist die evaluierte Konfiguration eindeutig an Hand der Versionsnummer identifiziert:

- *RNG.java, Version 1.21.2.3*

9 Ergebnisse der Evaluierung

Der zusammenfassende Prüfbericht [ETR] wurde gemäß den Schemaanforderungen, den Common Criteria [CC], der Methodologie [CEM] und den Anwendungshinweisen und Interpretationen des Schemas [AIS] von der Prüfstelle der *datenschutz nord* angefertigt.

Der EVG erfüllt alle Vertrauenswürdigkeitsanforderungen nach EAL 3 mit Zusatz ADV_LLD.1, ADV_IMP.1 und ALC_TAT.1. Die Ergebnisse zu den einzelnen Vertrauenswürdigkeitskomponenten sind in der nachstehenden Tabelle zusammengefasst:

Vertrauenswürdigkeitsklassen und -komponenten		**Urteil**
Sicherheitsvorgaben	**CC Klasse ASE**	ERFÜLLT
EVG-Beschreibung	ASE_DES.1	ERFÜLLT
Sicherheitsumgebung	ASE_ENV.1	ERFÜLLT
ST-Einführung	ASE_INT.1	ERFÜLLT
Sicherheitsziele	ASE_OBJ.1	ERFÜLLT
PP-Postulate	ASE_PPC.1	n. a.[4]
IT-Sicherheitsanforderungen	ASE_REQ.1	ERFÜLLT
Explizit dargelegte IT-Sicherheitsanforderungen	ASE_SRE.1	ERFÜLLT
EVG-Übersichtsspezifikation	ASE_TSS.1	ERFÜLLT
Konfigurationsmanagement	**CC Klasse ACM**	ERFÜLLT
Autorisierungskontrolle	ACM_CAP.3	ERFÜLLT
EVG-CM-Umfang	ACM_SCP.1	ERFÜLLT
Auslieferung und Betrieb	**CC Klasse ADO**	ERFÜLLT
Auslieferungsprozeduren	ADO_DEL.1	ERFÜLLT
Installations-, Generierungs- und Anlaufprozeduren	ADO_IGS.1	ERFÜLLT

[4] n. a. = nicht anwendbar

Entwicklung	**CC Klasse ADV**	ERFÜLLT
Informelle funktionale Spezifikation	ADV_FSP.1	ERFÜLLT
Sicherheitsspezifischer Entwurf auf hoher Ebene	ADV_HLD.2	ERFÜLLT
Beschreibender Entwurf auf niedriger Ebene	ADV_LLD.1	ERFÜLLT
Teilmenge der Implementierung der TSF	ADV_IMP.1	ERFÜLLT
Informeller Nachweis der Übereinstimmung	ADV_RCR.1	ERFÜLLT
Handbücher	**CC Klasse AGD**	ERFÜLLT
Systemverwalterhandbuch	AGD_ADM.1	ERFÜLLT
Benutzerhandbuch	AGD_USR.1	ERFÜLLT
Lebenszyklus-Unterstützung	**CC Klasse ALC**	ERFÜLLT
Identifikation der Sicherheitsmaßnahmen	ALC_DVS.1	ERFÜLLT
Klar festgelegte Entwicklungswerkzeuge	ALC_TAT.1	ERFÜLLT
Testen	**CC Klasse ATE**	ERFÜLLT
Analyse der Testabdeckung	ATE_COV.2	ERFÜLLT
Testen - Entwurf auf hoher Ebene	ATE_DPT.1	ERFÜLLT
Funktionales Testen	ATE_FUN.1	ERFÜLLT
Unabhängiges Testen – Stichprobenartig	ATE_IND.2	ERFÜLLT
Schwachstellenbewertung	**CC Klasse AVA**	ERFÜLLT
Prüfung der Handbücher	AVA_MSU.1	ERFÜLLT
Stärke der EVG-Sicherheitsfunktionen	AVA_SOF.1	ERFÜLLT
Schwachstellenanalyse des Entwicklers	AVA_VLA.1	ERFÜLLT

Die Sicherheitsvorgaben weisen aus, dass sie nicht auf einem Schutzprofil (Protection Profile = PP) beruhen, so dass die Komponente ASE_PPC.1 nicht anwendbar ist. Bei allen anderen Vertrauenswürdigkeitskomponenten wurde das Urteil „erfüllt“ vergeben. Dies bezieht sich auch auf alle Evaluator-Aktionselemente, die Teil der Vertrauenswürdigkeits-komponenten sind. Daher ist der EVG, wie in den Sicherheitsvorgaben beschrieben, CC Teil 3 konform.

Die Sicherheitsvorgaben (Kapitel 5) geben vor, dass der EVG die folgende IT-Sicherheitsanforderung erfüllt, die nicht aus [CC] Teil 2 sondern der [AIS 31] entnommen wurde:

Komponenten ID	Komponenten Titel
FCS_RND.1	Qualitätsmetrik für Zufallszahlen

Die Evaluation, die auf der Stufe EAL 3 mit Zusatz ADV_LLD.1, ADV_IMP.1 und ALC_TAT.1 durchgeführt wurde, zeigt, dass diese Sicherheitsanforderung des EVG durch die Sicherheitsfunktion korrekt umgesetzt und somit die Sicherheitsziele, wie sie in den Sicherheitsvorgaben vorgegeben sind, erfüllt wurden.

Sicherheitsfunktion SF1 erfüllt die Stärke der Funktionen “SOF-hoch”. Der Zufallszahlengenerator von SF1 erfüllt die Anforderungen der Klasse K3 mit SOF-hoch der [AIS 20]. Für die Beurteilung nach AIS 20 wurde im Rahmen der Erst-Zertifizierung vom Evaluator der gesamte Quellcode zum Zufallszahlengenerator untersucht.

Jede Änderung des EVG oder seiner Systemverwalter-/Benutzerdokumentation seitens des EVG-Herstellers ist der Prüfstelle und der Zertifizierungsstelle anzuzeigen und zieht ggf. eine Re-Evaluation bzw. Re-Zertifizierung nach sich. Die Ergebnisse dieser Evaluation sind nur auf das Produkt "*RNG.java, Version 1.21.2.3*" anwendbar. Die Gültigkeit der Ergebnisse kann auf neue Versionen des Produkts ausgeweitet werden, wenn der Antragsteller die Modifikationen zur Re-Evaluation und Re-Zertifizierung einreicht und die Evaluation keine Sicherheitsbedenken ausweist.

10 Auflagen, Empfehlungen und Kommentare der Evaluatoren

Der Evaluierungsbericht – Evaluation Technical Report (ETR) – enthält keine Auflagen. Die im ETR enthaltenen und die Evaluierungsdokumentation betreffenden Hinweise und Empfehlungen sind im Falle einer Re-Zertifizierung zu beachten.

11 Auflagen und Hinweise der Zertifizierung

Es gibt keine Auflagen aus diesem Zertifizierungsbericht.

Hinweis: Die Erst-Zertifizierung (TUVIT-TSZ-CC-9259-2007) ist weiterhin gültig. Daher kann neben der Sun JDK-Version 1.5.0_10 auch die Version 1.4.2_08 eingesetzt werden.

12 Sicherheitsvorgaben

Die Sicherheitsvorgaben [ST] zu "*RNG.java, Version 1.21.2.3*" sind im Teil D dieses Zertifizierungsberichts enthalten.

13 Definitionen

13.1 Abkürzungen

BD	Benutzerdokumentation
CC	Common Criteria for Information Technology Security Evaluation (referenced to as [CC])
CEM	Common Methodology for Information Technology Security Evaluation (referenced to as [CEM])
CM	Configuration Management (Konfigurationsmanagement)

DRNG	Deterministic Random Number Generator (Deterministischer Zufallszahlengenerator)
EAL	Evaluation Assurance Level (Vertrauenswürdigkeitsstufe)
EVG	Evaluationsgegenstand
FSP	Functional Specification (Funktionale Spezifikation)
HLD	High-level Design (Entwurf auf hoher Ebene)
IF	Interface (Schnittstelle)
OSP	Organisational Security Policy (Organisatorische Sicherheitspolitik)
PP	Protection Profile (Schutzprofil)
SAR	Security Assurance Requirement (Vertrauenswürdigkeitsanforderung)
SD	Systemverwalterdokumention
SF	Security Function (Sicherheitsfunktion)
SFP	Security Function Policy (Sicherheitspolitik)
SFR	Security Functional Requirement (Sicherheitsanforderungen)
SIF	Sub-interface
SOF	Strength of Function (Stärke der Funktionen)
SS	Subsystem
ST	Security Target (Sicherheitsvorgaben)
TSC	TSF Scope of Control (Anwendungsbereich der TSF-Kontrolle)
TSF	TOE Security Functions (EVG Sicherheitsfunktionen)
TSFI	TOE Security Function Interfaces (EVG SF Schnittstellen)
TSP	TOE Security Policy (EVG Sicherheitspolitik)
VLA	Vulnerability Analysis (Schwachstellenbewertung)

13.2 Glossar

Zusatz – Das Hinzufügen einer oder mehrerer Vertrauenswürdigkeitskomponenten aus Teil 3 der CC zu einer EAL oder einem Vertrauenswürdigkeitspaket.

Erweiterung – Das Hinzufügen von funktionalen Anforderungen, die nicht in Teil 2 enthalten sind, und/oder von Vertrauenswürdigkeitsanforderungen, die nicht in Teil 3 enthalten sind, zu den Sicherheitsvorgaben bzw. dem Schutzprofil.

Formal – Ausgedrückt in einer Sprache mit beschränkter Syntax und festgelegter Semantik, die auf bewährten mathematischen Konzepten basiert.

Informell – Ausgedrückt in natürlicher Sprache.

Objekt – Eine Einheit im TSC, die Informationen enthält oder empfängt und mit der Subjekte Operationen ausführen.

Schutzprofil – Eine implementierungsunabhängige Menge von Sicherheitsanforderungen für eine Kategorie von EVG, die besondere Konsumentenbedürfnisse erfüllen.

Sicherheitsfunktion – Ein Teil oder Teile eines EVG, auf die zur Durchsetzung einer hierzu in enger Beziehung stehenden Teilmenge der Regeln der EVG-Sicherheitspolitik Verlass sein muss.

Sicherheitsvorgaben – Eine Menge von Sicherheitsanforderungen und Sicherheitsspezifikationen, die als Grundlage für die Prüfung und Bewertung eines angegebenen EVG dienen.

Semiformal – Ausgedrückt in einer Sprache mit beschränkter Syntax und festgelegter Semantik.

Stärke der Funktionen – Eine Charakterisierung einer EVG-Sicherheitsfunktion, die den geringsten angenommenen Aufwand beschreibt, der notwendig ist, um deren erwartetes Sicherheitsverhalten durch einen direkten Angriff auf die zugrunde liegenden Sicherheitsmechanismen außer Kraft zu setzen.

SOF-Niedrig – Eine Stufe der EVG-Stärke von Funktionen, bei der die Analyse zeigt, dass die Funktionen einen angemessenen Schutz gegen zufälliges Brechen der EVG-Sicherheit durch Angreifer bieten, die über ein geringes Angriffspotential verfügen.

SOF-Mittel – Eine Stufe der EVG-Stärke von Funktionen, bei der die Analyse zeigt, dass die Funktionen einen angemessenen Schutz gegen naheliegendes oder absichtliches Brechen der EVG-Sicherheit durch Angreifer bieten, die über ein mittleres Angriffspotential verfügen.

SOF-Hoch – Eine Stufe der EVG-Stärke von Funktionen, bei der die Analyse zeigt, dass die Funktionen einen geeigneten Schutz gegen geplantes oder organisiertes Brechen der EVG-Sicherheit durch Angreifer bieten, die über ein hohes Angriffspotential verfügen.

Subjekt – Eine Einheit innerhalb des TSC, die die Ausführung von Operationen bewirkt.

Evaluationsgegenstand – Ein IT-Produkt oder –System – sowie die dazugehörigen Systemverwalter- und Benutzerhandbücher – das Gegenstand einer Prüfung und Bewertung ist.

EVG-Sicherheitsfunktionen – Eine Menge, die die gesamte Hardware, Software, und Firmware des EVG umfasst, auf die Verlass sein muss, um die TSP korrekt zu erfüllen.

EVG-Sicherheitspolitik – Eine Menge von Regeln, die angibt, wie innerhalb eines EVG Werte verwaltet, geschützt und verteilt werden.

Anwendungsbereich der TSF-Kontrolle – Die Menge der Interaktionen, die mit oder innerhalb eines EVG vorkommen können und den Regeln der TSP unterliegen.

14 Literaturangaben

[AIS] Anwendungshinweise und Interpretationen zum Schema (AIS)

[AIS 20] Anwendungshinweise und Interpretationen zum Schema (AIS) – Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, AIS 20, Version 1, 02.12.1999, Hrsg. BSI

[AIS 31] Anwendungshinweise und Interpretationen zum Schema (AIS) – Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, AIS 31, Version 1, 25.09.2001, Hrsg. BSI

[BD] Benutzerhandbuch – Casinoland-RNG (Zufallszahlengenerator RNG.java), Version 1.7, 19.12.2007, Autor: Casinoland GmbH

[CC] ISO/IEC 15408, Information technology – Security techniques – Evaluation criteria for IT security,
ISO/IEC 15408-1:2005 (E), Part 1: Introduction and general model
ISO/IEC 15408-2:2005 (E), Part 2: Security functional requirements
ISO/IEC 15408-3:2005 (E), Part 3: Security assurance requirements

[CEM] Common Methodology for Information Technology Security Evaluation,
Part 1: Introduction and general model, version 0.6, revision 11.01.1997,
Part 2: Evaluation Methodology, Version 2.3, August 2005

[ETR] Evaluation Technical Report, *datenschutz nord GmbH*,
version *0.3*, *12.01.2007*, project-number:
Casinoland_ETR_v03_20070112.doc

[MT 19937] Matsumoto, Makoto and Takuji Nishimura: „Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator”, ACM Trans. Model. Comput. Simul. 8, No.1, pp. 3-30 (1998).

[SD] Systemverwalterhandbuch – Casinoland-RNG (Zufallszahlengenerator RNG.java), Version 1.7, 19.12.2007, Autor: Casinoland GmbH

[ST] Sicherheitsvorgaben für den Zufallszahlengenerator des Online-Casinos – RNG.java, Version 1.9, 20.12.2007, Autor: Casinoland GmbH

Teil C

Auszüge aus den technischen Regelwerken (in Englisch)

Die Auszüge aus den Regelwerken umfassen die Themen:

- Kennzeichnung der Evaluationsergebnisse
- Vertrauenswürdigkeitskategorisierung
- Vertrauenswürdigkeitsstufen
- Stärke der Sicherheitsfunktionen
- Schwachstellenanalyse

CC Teil 1:

Conformance results

„The conformance result indicates the source of the collection of requirements that is met by a TOE or PP that passes its evaluation. This conformance result is presented with respect to Part 2 (functional requirements), Part 3 (assurance requirements) and, if applicable, to a pre-defined set of requirements (e.g., EAL, Protection Profile).

The conformance result consists of one of the following:

> **Part 2 conformant** - A PP or TOE is Part 2 conformant if the functional requirements are based only upon functional components in Part 2.
>
> **Part 2 extended** - A PP or TOE is Part 2 extended if the functional requirements include functional components not in Part 2.

plus one of the following:

> **Part 3 conformant** - A PP or TOE is Part 3 conformant if the assurance requirements are based only upon assurance components in Part 3.
>
> **Part 3 extended** - A PP or TOE is Part 3 extended if the assurance requirements include assurance requirements not in Part 3.

Additionally, the conformance result may include a statement made with respect to sets of defined requirements, in which case it consists of one of the following:

> ***Package name* Conformant** - A PP or TOE is conformant to a pre-defined named functional and/or assurance package (e.g. EAL) if the requirements (functions or assurance) include all components in the packages listed as part of the conformance result.
>
> ***Package name* Augmented** - A PP or TOE is an augmentation of a predefined named functional and/or assurance package (e.g. EAL) if the requirements (functions or assurance) are a proper superset of all components in the packages listed as part of the conformance result.

Finally, the conformance result may also include a statement made with respect to Protection Profiles, in which case it includes the following:

> ***PP Conformant*** - A TOE meets specific PP(s), which are listed as part of the conformance result."

CC Teil 3:

Assurance categorisation

The assurance classes, families, and the abbreviation for each family are shown in *Table 1*.

Assurance Class	Assurance Family	Abbreviated Name
Class ACM: Configuration management	CM automation	ACM_AUT
	CM capabilities	ACM_CAP
	CM scope	ACM_SCP
Class ADO: Delivery and operation	Delivery	ADO_DEL
	Installation, generation and start-up	ADO_IGS
Class ADV: Development	Functional specification	ADV_FSP
	High-level design	ADV_HLD
	Implementation representation	ADV_IMP
	TSF internals	ADV_INT
	Low-level design	ADV_LLD
	Representation correspondence	ADV_RCR
	Security policy modeling	ADV_SPM
Class AGD: Guidance documents	Administrator guidance	AGD_ADM
	User guidance	AGD_USR
Class ALC: Life cycle support	Development security	ALC_DVS
	Flaw remediation	ALC_FLR
	Life cycle definition	ALC_LCD
	Tools and techniques	ALC_TAT
Class ATE: Tests	Coverage	ATE_COV
	Depth	ATE_DPT
	Functional tests	ATE_FUN
	Independent testing	ATE_IND
Class AVA: Vulnerability assessment	Covert channel analysis	AVA_CCA
	Misuse	AVA_MSU
	Strength of TOE security functions	AVA_SOF
	Vulnerability analysis	AVA_VLA

Table 1: Assurance family breakdown and mapping

Evaluation assurance levels (chapter 6)

“The Evaluation Assurance Levels (EALs) provide an increasing scale that balances the level of assurance obtained with the cost and feasibility of acquiring that degree of assurance. The CC approach identifies the separate concepts of assurance in a TOE at the end of the evaluation, and of maintenance of that assurance during the operational use of the TOE.

It is important to note that not all families and components from Part 3 are included in the EALs. This is not to say that these do not provide meaningful and desirable assurances.

Instead, it is expected that these families and components will be considered for augmentation of an EAL in those PPs and STs for which they provide utility."

Evaluation assurance level (EAL) overview

„Table 2 represents a summary of the EALs. The columns represent a hierarchically ordered set of EALs, while the rows represent assurance families. Each number in the resulting matrix identifies a specific assurance component where applicable.

As outlined in the next section, seven hierarchically ordered evaluation assurance levels are defined in the CC for the rating of a TOE's assurance. They are hierarchically ordered inasmuch as each EAL represents more assurance than all lower EALs. The increase in assurance from EAL to EAL is accomplished by *substitution* of a hierarchically higher assurance component from the same assurance family (i.e. increasing rigour, scope, and/or depth) and from the *addition* of assurance components from other assurance families (i.e. adding new requirements).

These EALs consist of an appropriate combination of assurance components as described in chapter 2 of CC Part 3. More precisely, each EAL includes no more than one component of each assurance family and all assurance dependencies of every component are addressed.

While the EALs are defined in the CC, it is possible to represent other combinations of assurance. Specifically, the notion of "augmentation" allows the addition of assurance components (from assurance families not already included in the EAL) or the substitution of assurance components (with another hierarchically higher assurance component in the same assurance family) to an EAL. Of the assurance constructs defined in the CC, only EALs may be augmented. The notion of an "EAL minus a constituent assurance component" is not recognised by the CC as a valid claim. Augmentation carries with it the obligation on the part of the claimant to justify the utility and added value of the added assurance component to the EAL. An EAL may also be extended with explicitly stated assurance requirements."

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration Management	ACM_AUT				**1**	1	**2**	2
	ACM_CAP	**1**	**2**	**3**	**4**	4	**5**	5
	ACM_SCP			**1**	**2**	**3**	3	3
Delivery and Operation	ADO_DEL		**1**	1	**2**	2	2	**3**
	ADO_IGS	**1**	1	1	1	1	1	1
Development	ADV_FSP	**1**	1	1	**2**	**3**	3	**4**
	ADV_HLD		**1**	**2**	2	**3**	**4**	**5**
	ADV_IMP				**1**	**2**	**3**	3
	ADV_IMT					**1**	**2**	***3***
	ADV_LLD				**1**	1	**2**	2
	ADV_RCR	**1**	1	1	1	**2**	2	**3**
	ADV_SPM				**1**	**3**	3	3
Guidance Documents	AGD_ADM	**1**	1	1	1	1	1	1
	AGD_USR	**1**	1	1	1	1	1	1
Life Cycle Support	ALC_DVS			**1**	1	1	**2**	2
	ALC_FLR							
	ALC_LCD				**1**	**2**	2	**3**
	ALC_TAT				**1**	**2**	**3**	3
Tests	ATE_COV		**1**	**2**	2	2	**3**	3
	ATE_DPT			**1**	1	**2**	2	**3**
	ATE_FUN		**1**	1	1	1	**2**	2
	ATE_IND	**1**	**2**	2	2	2	2	**3**
Vulnerability Assessment	AVA_CCA					**1**	**2**	2
	AVA_MSU			**1**	**2**	2	**3**	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	**2**	**3**	**4**	4

Table 2: Evaluation assurance level summary

Evaluation assurance level 1 (EAL1) - functionally tested

“EAL1 is applicable where some confidence in correct operation is required, but the threats to security are not viewed as serious. It will be of value where independent assurance is required to support the contention that due care has been exercised with respect to the protection of personal or similar information.

EAL1 provides an evaluation of the TOE as made available to the customer, including independent testing against a specification, and an examination of the guidance documentation provided. It is intended that an EAL1 evaluation could be successfully conducted without assistance from the developer of the TOE, and for minimal outlay. An evaluation at this level should provide evidence that the TOE functions in a manner consistent with its documentation, and that it provides useful protection against identified threats.“

Evaluation assurance level 2 (EAL2) - structurally tested

“EAL2 requires the co-operation of the developer in terms of the delivery of design information and test results, but should not demand more effort on the part of the developer than is consistent with good commercial practice. As such it should not require a substantially increased investment of cost or time.

EAL2 is therefore applicable in those circumstances where developers or users require a low to moderate level of independently assured security in the absence of ready availability of the complete development record. Such a situation may arise when securing legacy systems, or where access to the developer may be limited.“

Evaluation assurance level 3 (EAL3) - methodically tested and checked

“EAL3 permits a conscientious developer to gain maximum assurance from positive security engineering at the design stage without substantial alteration of existing sound development practices.

EAL3 is applicable in those circumstances where developers or users require a moderate level of independently assured security, and require a thorough investigation of the TOE and its development without substantial re-engineering.“

Evaluation assurance level 4 (EAL4) - methodically designed, tested, and reviewed

“EAL4 permits a developer to gain maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. EAL4 is the highest level at which it is likely to be economically feasible to retrofit to an existing product line.

EAL4 is therefore applicable in those circumstances where developers or users require a moderate to high level of independently assured security in conventional commodity TOEs and are prepared to incur additional security-specific engineering costs.“

Evaluation assurance level 5 (EAL5) - semiformally designed and tested

“EAL5 permits a developer to gain maximum assurance from security engineering based upon rigorous commercial development practices supported by moderate application of specialist security engineering techniques. Such a TOE will probably be designed and developed with the intent of achieving EAL5 assurance. It is likely that the additional costs attributable to the EAL5 requirements, relative to rigorous development without the application of specialised techniques, will not be large.

EAL5 is therefore applicable in those circumstances where developers or users require a high level of independently assured security in a planned development and require a rigorous development approach without incurring unreasonable costs attributable to specialist security engineering techniques.“

Evaluation assurance level 6 (EAL6) - semiformally verified design and tested

“EAL6 permits developers to gain high assurance from application of security engineering techniques to a rigorous development environment in order to produce a premium TOE for protecting high value assets against significant risks.

EAL6 is therefore applicable to the development of security TOEs for application in high risk situations where the value of the protected assets justifies the additional costs.“

Evaluation assurance level 7 (EAL7) - formally verified design and tested

“EAL7 is applicable to the development of security TOEs for application in extremely high risk situations and/or where the high value of the assets justifies the higher costs. Practical application of EAL7 is currently limited to TOEs with tightly focused security functionality that is amenable to extensive formal analysis.“

Strength of TOE security functions (AVA_SOF)

AVA_SOF Strength of TOE security functions

“Strength of function analysis addresses TOE security functions that are realised by a probabilistic or permutational mechanism (e.g. a password or hash function). Even if such functions cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat them by direct attack. A level or a specific metric may be claimed for the strength of each of these functions. Strength of function analysis is performed to determine whether such functions meet or exceed the claim. For example, strength of function analysis of a password mechanism can demonstrate that the password function meets the strength claim by showing that the password space is sufficiently large.”

Three levels exist: SOF-basic, SOF-medium, and SOF-high.

Vulnerability analysis (AVA_VLA)

AVA_VLA Vulnerability analysis

"Vulnerability analysis consists of the identification of flaws potentially introduced in the different refinement steps of the development. It results in the definition of penetration tests through the collection of the necessary information concerning: (1) the completeness of the TSF (does the TSF counter all the postulated threats?) and (2) the dependencies between all security functions. These potential vulnerabilities are assessed through penetration testing to determine whether they could, in practice, be exploitable to compromise the security of the TOE."

Application notes

"A vulnerability analysis is performed by the developer in order to ascertain the presence of security vulnerabilities, and should consider at least the contents of all the TOE deliverables including the ST for the targeted evaluation assurance level. The developer is required to document the disposition of identified vulnerabilities to allow the evaluator to make use of that information if it is found useful as a support for the evaluator's independent vulnerability analysis.

The intent of the developer analysis is to confirm that no identified security vulnerabilities can be exploited in the intended environment for the TOE and that the TOE is resistant to obvious penetration attacks.

Obvious vulnerabilities are considered to be those that are open to exploitation that requires a minimum of understanding of the TOE, skill, technical sophistication, and resources. These might be suggested by the TSF interface description. Obvious vulnerabilities include those in the public domain, details of which should be known to a developer or available from an evaluation authority.

Performing a search for vulnerabilities in a systematic way requires that the developer identify those vulnerabilities in a structured and repeatable way, as opposed to identifying them in an ad-hoc fashion. The associated evidence that the search for vulnerabilities was systematic should include identification of all TOE documentation upon which the search for flaws was based.

Independent vulnerability analysis goes beyond the vulnerabilities identified by the developer. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low (for AVA_VLA.2), moderate (for AVA_VLA.3) or high (for AVA_VLA.4) attack potential. To accomplish this intent, the evaluator first assesses the exploitability of all identified vulnerabilities. This is accomplished by conducting penetration testing. The evaluator should assume the role of an attacker with a low (for AVA_VLA.2), moderate (for

AVA_VLA.3) or high (for AVA_VLA.4) attack potential when attempting to penetrate the TOE. Any exploitation of vulnerabilities by such an attacker should be considered by the evaluator to be “obvious penetration attacks” (with respect to the AVA_VLA.*.2C elements) in the context of the components AVA_VLA.2 through AVA_VLA.4.”

Teil D

Sicherheitsvorgaben

Es folgen die Sicherheitsvorgaben: "*Sicherheitsvorgaben für den Zufallszahlengenerator des Online-Casinos – RNG.java*"

Author: Casinoland GmbH

Date: 20.12.2007

Version: 1.9

Sicherheitsvorgaben für den Zufallszahlengenerator des Online-Casinos

RNG.java

Casinoland GmbH

Version 1.9

20.12.2007

Historie

Version	Datum	geänderte Kapitel	Grund der Änderung	Geändert durch
1.0	12.09.2006		Erstellung	datenschutz nord GmbH, Ralf von Rahden
1.1	04.10.2006	Kapitel 1.1 und 1.2 Kapitel 2	Aktualisierung der Versionsnummer Ergänzung des Auslieferungsumfang	datenschutz nord GmbH, Ralf von Rahden
1.2	10.10.2006	1.1	Ergänzungen, Aktualisierungen	agens PSC GmbH, Corinna Brunstein
1.3	01.11.2006	Kapitel 1.2 Kapitel 1.3 Kapitel 2 Kapitel 3.1 Kapitel 5.1.1 Kapitel 5.2 Kapitel 8.1 Kapitel 8.2	Überarbeitung Überarbeitung EVG-Beschreibung: Herausnahme von Details Detailliertere Beschreibung der Annahmen Definition aus [AIS 31] Ergänzung der Anforderungen an die IT-Umgebung Anpassen der Erklärungen an die veränderten Annahmen Anpassen an die veränderte funktionale Anforderung	datenschutz nord GmbH, Ralf von Rahden
1.4	22.11.2006	div.	Überarbeitung Versionen, Wertebereich, Referenzen	agens PSC GmbH, Corinna Brunstein
1.5	28.11.2006	Kapitel 1.2	Änderungen aus onsite visit Wertebereich geändert	agens PSC GmbH, Corinna Brunstein
		Kapitel 2	Formulierungen hins. Produkt	
		Kapitel 3.1	Annahmen überarbeitet	
		Kapitel 4	Sicherheitsziele den überarbeiteten Annahmen angepasst	
		Kapitel 6	Formulierungen hins. Produkt	
		Kapitel 8.1	Erklärung der Sicherheitsziele den geänderten Annahmen angepasst	
		div.	Löschen Funktionalitätsklasse ADV_SPM.1	
1.6	08.01.2007	Kapitel 2	Bereitstellung SHA-512	agens PSC GmbH, Sven Rill
		Kapitel 3.1	A.5, A.6	
		Kapitel 4.2	OE.5, OE.6	

		Kapitel 5.2	Ausführung der Operation ergänzt	
		Kapitel 5.2, 8.2.1 und 8.2.3	Ergänzung bei der Assignment Operation „cryptographic key size“	
		Kapitel 5.2	Ergänzung FDP_ITC.2	
		Kapitel 6.1	Erläuterung transition function	
1.7	09.01.2007	div	Aktualisierung der Versionsnummer	agens PSC GmbH, Sven Rill
1.8	10.01.2007	Kapitel 2.1	Wertebereich	agens PSC GmbH, Sven Rill
		Kapitel 5.2, 8.2.1 und 8.2.3	Anpassung „cryptographic key size none“	
		Kapitel 6.1 und 8.3.1	Überarbeitung SF1	
1.9	20.12.2007	Kapitel 3.1	Änderung der Einsatzumgebung von JDK 1.4.2 auf 1.5.0	agens PSC GmbH, Corinna Brunstein
		Kapitel 4.2		

Dokumenten-Überwachungsverfahren:

Status: final	Prozess-/Dokumentbesitzer: Corinna Brunstein, agensPSC GmbH Sven Rill, agensPSC GmbH Frank Neubauer, Spielbanken Niedersachsen

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 ST-Einführung

1.1 ST-Identifikation

1 ST-Name: Sicherheitsvorgaben für den Zufallszahlengenerator des Online-Casinos der Casinoland GmbH

2 ST-Version: 1.9

3 Datum: 20.12.2007

4 Autoren: Casinoland GmbH, datenschutz nord GmbH

5 TOE-Name: RNG.java

6 TOE-Version: 1.21.2.3

7 CC-Version: 2.3

1.2 ST-Übersicht

8 Der Evaluationsgegenstand (TOE = Target of Evaluation) und Gegenstand dieser Sicherheitsvorgaben ist die in Java realisierte Implementation in der Version 1.21.2.3 eines deterministischen Zufallszahlengenerators, der z. B. auf Online-Spielsystemen eingesetzt werden kann.

9 Deterministische Zufallszahlengeneratoren sind Algorithmen, die nach Eingabe eines Startwertes (Seed) eine Folge von Zahlen generiert. Diese Zahlen sollen für einen neutralen Betrachter die gleichen Eigenschaften haben wie echte Zufallszahlen. Diese lassen sich durch bestimmte statistische Tests überprüfen. Man spricht von den erzeugten Zahlen auch von Pseudozufallszahlen, da jemand, der den Seed und den Algorithmus kennt, in der Lage ist, dieselbe Zahlenfolge zu reproduzieren.

10 Der TOE erzeugt auf einem Linuxsystem auf deterministische Weise Zahlen zwischen 0 und 2^{31}-2. Da es sich bei einem Einsatz z. B. bei einer Spiele-Software möglicherweise um Glücksspiele mit Geldeinsatz handeln kann, müssen an die erzeugten Pseudozufallszahlen (im Folgenden Zufallzahlen genannt) hohe Anforderungen an die Qualität und Sicherheit gestellt werden.

11 Für die Generierung des Seed des TOE werden echte Zufallszahlen verwendet, so dass es selbst bei Kenntnis des Algorithmus nicht möglich ist, dieselbe Zahlenfolge zu erzeugen wie der TOE. Darüber hinaus ist es auf Grund des Designs des TOE nicht möglich, selbst bei Kenntnis beliebig vieler Ausgaben des TOE, eine weitere Ausgabe mit einer Wahrscheinlichkeit signifikant größer als $1/2^{32}$ vorherzusagen. Die durchgeführten und dokumentierten Herstellertests zeigen, dass die vom TOE erzeugten Zufallszahlen den Anforderungen der statistischen Tests, die in [AIS 20, Anhang F] beschriebenen sind, genügen.

12 Die Sicherheitsvorgaben stellen die funktionalen sowie organisatorischen Sicherheitsanforderungen und -prozeduren an den TOE und dessen Einsatzumgebung dar, die gemäß den Sicherheitszielen gewählt wurden.

1.3 Postulat der Übereinstimmung mit den Common Criteria

13 Der in Abschnitt 2 beschriebene Evaluationsgegenstand wird konform zu folgenden Teilen der Common Criteria entwickelt:

- Teil 1 [CC-Teil1]
- Teil 2 erweitert [CC-Teil2]
- Teil 3 mit Zusatz, EAL3 [CC-Teil3] mit den Zusätzen ADV_IMP.1, ALC_TAT.1, ADV_LLD.1 (abkürzend als EAL3+ bezeichnet).

14 Die Generierung von Zufallszahlen ist in den CC, Teil 2, Version 2.3, noch nicht als funktionale Anforderung beschrieben. Sie ist aber im Zusammenhang mit der Schlüsselgenerierung auf Smartcards [BSI PP0002] und für physikalische Zufallszahlengeneratoren in [AIS 31] als FCS_RND explizit dargelegt. Daher wird die zuletzt beschriebene Version dieser Klasse, die aus [AIS 31], in diesen Sicherheitsvorgaben verwendet. Hierbei wird dem Unterschied zu physikalischen Zufallszahlengeneratoren und zu Systemen, in denen die Zufallszahlen weiterverwendet werden, etwa für die Schlüsselgenerierung in einem Verschlüsselungssystem, Rechnung getragen, in der Form, dass die Komponente FCS_RND.1.2, siehe Absatz 84, sowie die Abhängigkeit zu FPT_TST.1 im Zusammenhang mit deterministischen Zufallszahlengeneratoren, wie in Abschnitt 8.2.3 beschrieben, nicht sinnvoll sind und daher nicht zur Anwendung kommen.

15 Hinsichtlich Teil 3 der CC soll der Zufallszahlengenerator die Vertrauenswürdigkeitsstufe EAL 3+ erreichen. Zusätzlich zu EAL 3 ist aus Sicht des BSI eine Quelltextanalyse (ADV_IMP.1) notwendig. Des Weiteren ist es aufgrund von Abhängigkeiten innerhalb der CC erforderlich, bei EAL 3+ einen beschreibenden Entwurf auf niedriger Ebene (ADV_LLD.1) bereitzustellen und eine Dokumentation der Entwicklungswerkzeuge (ALC_TAT.1) zu erstellen.

2 TOE – Beschreibung

16 Beim Evaluationsgegenstand handelt es sich um die Funktion zur Erzeugung von Pseudozufallszahlen, im Folgenden Zufallszahlengenerator (RNG = Random Number Generator) genannt. Die Funktion ist in Java-Code implementiert. Die Funktion des Zufallszahlengenerators ist in der kompilierten Version als eine eigene Java-Klasse abgegrenzt.

17 Der RNG wird gemäß [AIS 20] beschrieben durch das 5-Tupel (S,R,φ,ψ,p_A):

18 S die (endliche) Menge der möglichen inneren Zustände des Zufallszahlengenerators,

19 R die Menge der möglichen Ausgabewerte (Zufallszahlen),

20 $\varphi: S \rightarrow S$ Zustandsfunktion der inneren Zustände,

21 $\psi: S \rightarrow R$ Ausgabefunktion,

22 p_A ein Wahrscheinlichkeitsmaß, das die zufällige Verteilung des Anfangszustands $s_0 \in S$ beschreibt.

23 Des Weiteren ist eine obere Schranke M für die Anzahl von Zufallszahlen, die während eines gesamten Lebenszyklus maximal generiert werden, anzugeben.

24 Der TOE soll eine Zufallszahl in weniger als fünf Millisekunden ausgeben können. Unter der Annahme, dass jede Millisekunde eine Zufallszahl ausgegeben wird, würde der RNG am Tag 86400000, also etwa 2^{26} Zahlen erzeugen. Das hieße ungefähr 2^{31} im Monat, ca. 2^{34} im Jahr und etwa 2^{38} in zehn Jahren. Die obere Schranke M muss diesen Wert deutlich überschreiten, damit ein Neustart des TOE aus Sicherheitsgründen nicht notwendig wird. Daher soll für den TOE mindestens $M=2^{512}$ betragen. Sollte der TOE in einem Lebenszyklus nur eine kleine Anzahl Zahlen generiert haben, etwa auf Grund eines Systemausfalls, so entstünde hierdurch kein Sicherheitsrisiko. Bei jedem Start des TOE wird der RNG mit einem neuen Startwert (Seed) initialisiert, der eine Entropie von mindestens 80 besitzt. Das heißt, dass nur mit einer Wahrscheinlichkeit von höchstens $1/2^{80}$ dieselben Zahlen wieder erzeugt würden.

25 Der TOE besitzt zwei Hauptkomponenten, mit denen die gewünschten Eigenschaften erreicht werden. Zunächst werden Zufallszahlen mit Hilfe des Mersenne Twister 19937 [MT 19937] erzeugt. Diese stellen die inneren Zustände des RNG dar. Das stellt sicher, dass eine ausreichend große Periode erzeugt wird. Außerdem haben diese Zahlen bereits sehr gute statistische Eigenschaften, da sie die Diehard Tests [DIEHARD] laut [MT 19937] bestehen.

26 Anschließend, als Teil der Ausgabefunktion, wird auf diese Zahlen der Hash-Algorithmus SHA-512 [NIST 180-2] angewendet. Da es sich hierbei um eine starke Einwegfunktion handelt, wird hierdurch erreicht, dass die Ausgaben des RNG nicht vorhersagbar sind. Der Hash-Algorithmus SHA-512 wird als Teil der EVG-Umgebung durch die Java Bibliothek „java.security.MessageDigest“ bereitgestellt.

27 Der Auslieferungsumfang des TOE umfasst das Programm RNG.java in der Version 1.21.2.3, das Administratoren- und Benutzerhandbuch.

2.1 Abgrenzung von der TOE-Umgebung

28 Der TOE ist eine in Java realisierte Funktionsbibliothek und kann in Java-Programmen wie z. B. Online-Spielsystemen auf einem sicheren Server eingesetzt werden, um Zufallszahlen zu erzeugen.

29 Der TOE wird als Java-Bibliothek durch „RNG.Randrange(N)“ aufgerufen. Beim ersten Aufruf wird der TOE initialisiert. Bei jedem Aufruf, also auch beim ersten, wird der innere Zustand neu berechnet, und aus diesem die Ausgabe des TOE, eine Zufallszahl im Bereich $[0,2^{31}-2]$.

30 Der TOE hat drei eindeutige Schnittstellen (vgl. Abbildung 1):

1. **Eingabe:** Der TOE wird mit einem Startwert (Seed) initialisiert. Der Seed ist ein Element des Zahlenraums S, der inneren Zustände des RNG, und kann als erster innerer Zustand angesehen werden. Die Seed-Generierung ist nicht Bestandteil des TOE. Die Umgebung des TOE stellt die geforderte Qualität des Seed sicher.

2. **Java Bibliothek:** Der TOE ruft die eingebundene Java Bibliothek „java.security.MessageDigest“ auf, um den SHA-512 Algorithmus einzubinden. Diese Bibliothek gehört nicht zum Umfang des TOE, sondern zur TOE-Umgebung und stellt den Hash-Algorithmus SHA-512 für den TOE bereit.
3. **Ausgabe:** Der Aufruf des TOE wird durch den Befehl “RNG.randrange(N)” kontrolliert. Daraufhin wird eine Zufallszahl aus dem Bereich $[-2^{31}, 2^{31}-1]$ erzeugt und an die transition-function, übergeben. Die transition-function berechnet eine Zahl aus dem Bereich $[0, 2^{31}-2]$ als Rückgabewert.

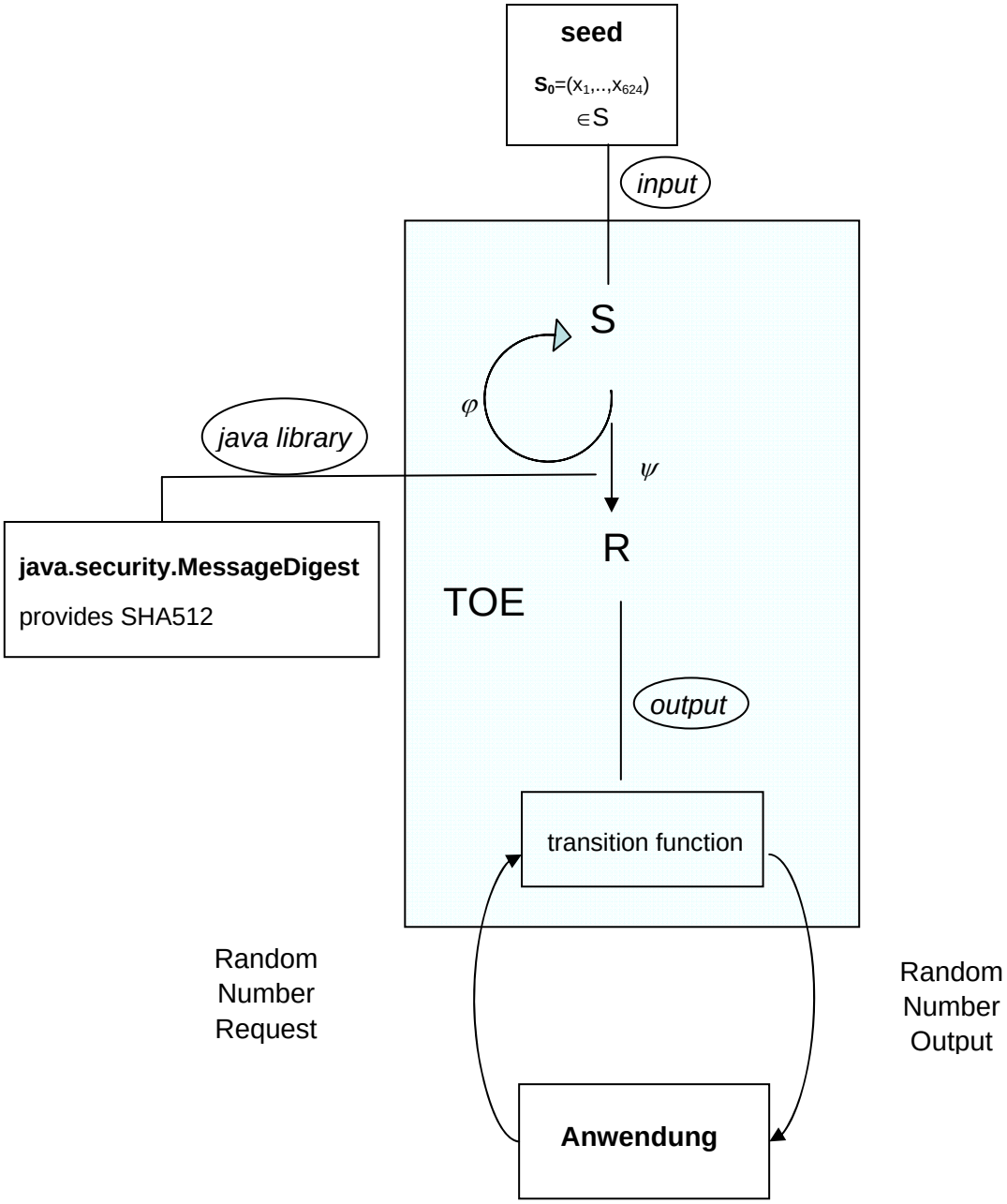

Abbildung 1: Schnittstellen des TOE

3 TOE – Sicherheitsumgebung

3.1 Annahmen

31 Der TOE stellt Zufallszahlen für Java-Programme bereit und besitzt keinen Einfluss auf die Sicherheit der externen Schnittstellen. Er wird auf einem System installiert, dass im Folgenden nur mit „System“ bezeichnet ist, und in der Beschreibung der funktionalen Spezifikation in [RNG-Develop] und in [Syste] präzisiert wird.

32 Damit die Integrität des TOE geschützt bleibt und die Sicherheit für seinen Einsatz gewährleistet ist, müssen entsprechend starke Annahmen an die Umgebung gestellt werden:

Tabelle 1: Annahmen an die Umgebung

A.1	Es wird angenommen, dass der TOE so betrieben wird, dass nur befugte Personen physikalischen Zugriff haben.
A.2	Der Zugriff auf das System und den TOE wird durch ein restriktives Berechtigungskonzept geschützt, so dass nur befugte Personen logischen Zugriff haben.
A.3	Es wird angenommen, dass die Integrität des Systems bei der Installation und während des Betriebs durch organisatorische Maßnahmen sichergestellt wird.
A.4	Es wird angenommen, dass bei der Installation sichergestellt wird, dass der zertifizierte TOE zum Einsatz kommt.
A.5	Es wird angenommen, dass der TOE mit einem seed von ausreichend hoher Entropie ($H(p_A) \geq 80$) initialisiert wird, welcher durch die Verwendung eines Redhat Linux-System mit Kernel-Version 2.6.9* in der Distribution Red Hat Linux ES 4 x86 bereitgestellt wird.
A.6	Es wird angenommen, dass die TOE-Umgebung den Hash-Algorithmus SHA-512 durch die Verwendung einer Java-VM Sun JDK-Version 1.5.0_10 bereitstellt.
A.7	Es wird angenommen, dass die Integrität des Systems inklusive der Java-Bibliothek „java.security.MessageDigest“ und des TOE regelmäßig überprüft wird.
A.8	Es wird angenommen, dass die eingesetzten Administratoren vertrauenswürdig sind.

3.2 Bedrohungen

33 Aufgrund der Annahmen hat ein Angreifer keinen direkten Zugriff auf das System, auf dem der TOE installiert ist. Als Grundlage für einen Angriff stehen einem Angreifer nur die Ausgaben des TOE zur Verfügung. Da ein er-

folgreicher Angriff bei einem Einsatz des TOE z. B. auf einem Online-Spielsystem einen großen finanziellen Gewinn für einen Angreifer bedeuten würde, ist davon auszugehen, dass ein Angreifer die zur Zeit verfügbaren technischen Möglichkeiten hat und bereit ist, einen großen zeitlichen Aufwand für einen Angriff zu betreiben.

34 Ziel eines Angreifers ist es, die Ausgaben des RNG vorherzubestimmen. Frühere Ausgaben des TOE stellen kein Geheimnis dar. Selbst wenn es möglich wäre, anhand von Ausgaben frühere Ausgaben zu berechnen, ist es aufgrund der großen Periodenlänge von mindestens 2^{512} technisch nicht möglich, die gesamte Ausgabe einer Periode zu berechnen. Bei einer Zahl pro Millisekunde würde man ungefähr 2^{478} Jahre benötigen. Aus diesem Grund stellen sie keine Bedrohung dar.

35 Die Java-Bibliothek, die die Hash-Funktion bereitstellt wird laut Annahme A.7 regelmäßig auf Integrität überprüft, so dass sich diese Schnittstelle nicht für einen Angriff ausnutzen lässt.

36 Es bleibt daher nur die Bedrohung T.1:

Tabelle 2: Bedrohungen

T.1	Ein Angreifer von außerhalb ist in der Lage auf Grund gesammelter Ausgaben des TOE die nächste Ausgabe vorherzusagen.

3.3 Organisatorische Sicherheitspolitik

37 Es sind keine organisatorischen Sicherheitspolitiken vorgesehen.

4 Sicherheitsziele

4.1 Sicherheitsziele für den TOE

38 Der TOE soll Zufallszahlen generieren, die möglichst gleichverteilt sind und geeignete statistische Tests bestehen. Die Zahlen sollen nicht vorhersagbar sein und eine hohe Entropie besitzen. Diese Ziele lassen sich zusammenfassen:

Tabelle 3: Sicherheitsziele für den TOE

O.1	Der TOE muss Zufallszahlen erzeugen, die der Funktionalitätsklasse K3 [AIS 20] mit Mechanismenstärke „hoch“ genügen.

4.2 Sicherheitsziele für die Umgebung

Tabelle 4: Sicherheitsziele für die Umgebung

OE.1	Zutrittskontrollen und Zugriffsbeschränkungen müssen den TOE vor physikalischem Zugriff von Unbefugten schützen.
OE.2	Es muss ein restriktives Berechtigungskonzept existieren, so dass nur befugte Personen logischen Zugriff haben.
OE.3	Die Integrität des Systems bei der Installation und während des Betriebes muss durch organisatorische Maßnahmen sichergestellt werden.
OE.4	Bei der Installation muss sichergestellt werden, dass der zertifizierte TOE zum Einsatz kommt.
OE.5	Der TOE muss mit einem seed von ausreichend hoher Entropie ($H(p_A) \geq 80$) unter Verwendung eines Redhat Linux-System mit Kernel-Version 2.6.9* in der Distribution Red Hat Linux ES 4 x86 initialisiert werden.
OE.6	Die TOE-Umgebung muss den Hash-Algorithmus SHA-512 unter Verwendung einer Java-VM Sun JDK-Version 1.5.0_10 bereitstellen.
OE.7	Die Integrität des Systems inklusive der Java-Bibliothek „java.security.MessageDigest“ und des TOE muss regelmäßig überprüft werden.
OE.8	Die Administratoren müssen vertrauenswürdig sein.

5 IT-Sicherheitsanforderungen

5.1 TOE-Sicherheitsanforderungen

5.1.1 Funktionale Sicherheitsanforderungen an den TOE

39 Die relevante funktionale Sicherheitsanforderung, FCS_RND.1[1], wird in [AIS 31] definiert. Es ist keine weitere notwendig, da die erzeugten Zufallszahlen nicht zur Erzeugung kryptographischer Schlüssel verwendet werden. Die in der Definition in [AIS 31] enthaltene Komponente FCS_RND.1.2, sowie die Abhängigkeit zu FPT_TST.1 gelten, wie in Abschnitt 8.2.3 erklärt, nur für physikalische und nicht für deterministische Zufallszahlengeneratoren.

40 Familie FCS_RND Erzeugung von Zufallszahlen

41 FCS_RND.1 Qualitätsmetrik für Zufallszahlen

42 FCS_RND.1.1 Die TSF müssen einen Mechanismus bereitstellen, um Zufallszahlen zu generieren, die **der Funktionalitätsklasse K3**

[1] Die Familie FCS_RND ist nicht in den CC Version 2.3 enthalten. Diese wird explizit in [AIS 31] dargelegt.

und Mechanismenstärke hoch, wie in [AIS 20] beschrieben[2] entsprechen.

43 FCS_RND.1.2 Die TSF müssen in der Lage sein, den Gebrauch der TSF-generierten Zufallszahlen für **keine TSF-Funktion[3]** durchzusetzen.

44 Abhängigkeiten: FPT_TST.1 TSF testing

5.1.2 Anforderungen an die Vertrauenswürdigkeit des TOE

45 Die Anforderungen an die Vertrauenswürdigkeit des TOE sollen über die Vertrauenswürdigkeitsstufe EAL3 hinausgehen. Die Komponenten, die benötigt werden, um die Stufe EAL3 zu erreichen, werden ergänzt um ADV_IMP.1, ADV_LLD.1, ALC_TAT.1 (abkürzend als EAL3+ bezeichnet).

Tabelle 5: Anforderungen an die Vertrauenswürdigkeit

Class	component	
Development (ADV)	ADV_FSP.1	Informal functional specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_LLD.1	Descriptive low-level design
	ADV_IMP.1	Subset of the implementation of the TSF
	ADV_RCR.1	Informal correspondence demonstration
Tests (ATE)	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional Tests
	ATE_IND.2	independent testing – sample
Configuration management (ACM)	ACM_CAP.3	Authorisation controls
	ACM_SCP.1	TOE CM coverage
Vulnarability assessment (AVA)	AVA_MSU.1	Examination of guidance
	AVA_SOF.1	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis
Life-cycle support (ALC)	ALC_DVS.1	Identification of security measures
	ALC_TAT.1	Well-defined development tools

[2] Operation: Zuweisung „definierte Qualitätsmetrik“.

[3] Operation: Zuweisung „Liste der TSF-Funktionen“.

Delivery and Operation (ADO)	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation und start-up procedures
Guidance documents (AGD)	AGD_ADM.1	administrator guidance
	AGD_USR.1	user guidance

5.2 Sicherheitsanforderungen an die IT-Umgebung

46 Es gibt zwei Anforderungen an die Sicherheit der IT-Umgebung. Zum einen die Generierung eines guten Seed und zum anderen die Bereitstellung der Hashfunktion SHA-512. Bei den funktionalen Anforderungen an die IT-Umgebung bezieht sich in diesem Kapitel die Bezeichnung „TSF“ sinngemäß auf die Sicherheitsfunktionen der IT-Umgebung.

47 Der Seed stellt für sich ebenfalls eine Zufallszahl dar. Daher ist diese Anforderung mit der Familie FCS_RND zu beschreiben:

48 Familie FCS_RND Erzeugung von Zufallszahlen

49 FCS_RND.1 Qualitätsmetrik für Zufallszahlen

50 FCS_RND.1.1 Die TSF müssen einen Mechanismus bereitstellen, um Zufallszahlen zu generieren, die **mindestens einer Entropie von 80**[4] entsprechen.

51 FCS_RND.1.2 Die TSF müssen in der Lage sein, den Gebrauch der TSF-generierten Zufallszahlen für **keine TSF-Funktion**[5] durchzusetzen.

52 Abhängigkeiten: FPT_TST.1 TSF testing

53

54 FPT_TST.1 TSF testing

55 Hierarchical to: No other components.

56 Dependencies: FPT_AMT.1 Abstract machine testing

57 FPT_TST.1.1 The TSF shall run a suite of self tests **periodically during normal operation**[6] to demonstrate the correct operation of **the TSF**[7]**.**

58 FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of **the TSF**[8].

[4] Operation: Zuweisung „definierte Qualitätsmetrik“.

[5] Operation: Zuweisung „Liste der TSF-Funktionen“.

[6] Operation: selection

[7] Operation: selection

[8] Operation: selection

59 FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

60 Dependencies: FPT_AMT.1 Abstract machine testing

61 FPT_AMT.1 Abstract machine testing

62 Hierarchical to: No other components.

63 FPT_AMT.1.1 The TSF shall run a suite of tests **periodically during normal operation**[9] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

64 Dependencies: No dependencies

65 Die Bereitstellung der Hashfunktion lässt sich in den CommonCriteria, Teil 2 [CC-Teil2] definieren mit der Familie FCS_COP:

66 FCS_COP.1 Cryptographic operation

67 Hierarchical to: No other components.

68 FCS_COP.1.1 The TSF shall perform **the computation of a hash-digest**[10] in accordance with a specified cryptographic algorithm **SHA-512**[11] and cryptographic key sizes **none** that meet the following: **Secure Hash Standard [NIST 180-2]**[12].

69 Dependencies: [FDP_ITC.1 Import of user data without security attributes

or

FDP_ITC.2 Import of user data with security attributes

or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

70 Im Kapitel Erklärungen wird gezeigt, dass die Abhängigkeiten von FCS_COP.1.1 in diesem Zusammenhang nicht relevant sind. Daher werden diese an dieser Stelle nicht vollständig angegeben, sondern nur ihre Namen.

[9] Operation: selection

[10] Operation: assignment: list of cryptographic operations

[11] Operation: assignment: cryptographic algorithm

[12] Operation: assignment: list of standards

6 TOE-Übersichtsspezifikation

6.1 TOE-Sicherheitsfunktionen

71 Der TOE generiert Zufallszahlen, wie in Abschnitt 2 dargestellt. Ließen sich die Ausgaben des TOE vorhersagen, so wäre die Grundfunktionalität, zufällige Zahlen zu liefern, nicht gegeben. Die Sicherheitsfunktionen des TOE müssen dies verhindern.

72 Diese Anforderung wird durch die Sicherheitsfunktion SF1 erfüllt:

SF1 Generierung von Zufallszahlen der Funktionalitätsklasse K3 gemäß [AIS 20] mit Mechanismenstärke „hoch“ incl. einer transition-function, die eine Zufallszahl aus dem Bereich [0, 2^{31}-2] liefert.

6.2 Maßnahmen zur Vertrauenswürdigkeit

73 Um die Vertrauenswürdigkeitsstufe EAL3+ zu erhalten, werden folgende Maßnahmen durchgeführt (vgl. Tabelle 6: Maßnahmen zur Erfüllung von EAL3+):

Tabelle 6: Maßnahmen zur Erfüllung von EAL3+

Anforderungen gemäß EAL3+		Maßnahmen der Entwickler
Auslieferung und Betrieb	ADO_DEL.1	Dokumentation und Anwendung von Prozeduren für die Auslieferung sowie die sichere Installation, Generierung und den sicheren Anlauf des TOE
	ADO_IGS.1	
Entwicklung	ADV_FSP.1	Definition von Anforderungen gemäß CC an die Entwicklungsprozeduren und Dokumentation
	ADV_HLD.2	
	ADV_LLD.1	
	ADV_IMP.1	
	ADV_RCR.1	
Handbücher	AGD_ADM.1	Erstellung und Auslieferung eines Systemverwalter- und eines Benutzerhandbuchs
	AGD_USR.1	
Konfigurationsmanagement	ACM_CAP.3	Verwendung eines Konfigurationsmanagements
	ACM_SCP.1	
Lebenszyklus-Unterstützung	ALC_DVS.1	Gewährleistung des Entwicklungsprozesses durch physikalische, personelle und sonstige Sicherheitsmaßnahmen
	ALC_TAT.1	
Testen	ATE_COV.2	Tests des Herstellers, Dokumentation der Ergebnisse sowie unabhängiges Testen durch den Evaluator
	ATE_DPT.1	
	ATE_FUN.1	

Anforderungen gemäß EAL3+		Maßnahmen der Entwickler
	ATE_IND.2	
Schwachstellenbewertung	AVA_MSU.1	Erstellung von Missbrauchsanalysen, Analyse für die sicherheitsrelevanten Mechanismen in Bezug auf die Mechanismenstärke, sowie Schwachstellenanalyse für alle offensichtlichen Schwachstellen des TOE
	AVA_SOF.1	
	AVA_VLA.1	

7 PP-Postulate

74 Es ist keine Konformität zu einem PP vorgesehen.

8 Erklärung

8.1 Erklärung der Sicherheitsziele

75 In diesem Kapitel wird aufgezeigt, dass die beschriebenen Sicherheitsziele geeignet sind, allen Bedrohungen und Annahmen zu begegnen.

76 In der nachfolgenden Tabelle wird für jedes Sicherheitsziel des TOE und der Umgebung angegeben, welche Bedrohungen abgewehrt und welche Annahmen berücksichtigt werden sollen.

Tabelle 7: Zuordnung: Bedrohungen/Annahmen - Sicherheitsziele

	O.1	**OE.1**	**OE.2**	**OE.3**	**OE.4**	**OE.5**	**OE.6**	**OE.7**	**OE.8**
T.1	X								
A.1		X							
A.2			X						
A.3				X					
A.4					X				
A.5						X			
A.6							X		
A.7								X	
A.8									X

77 Es ist leicht zu sehen, dass jede Annahme und jede Bedrohung durch ein Sicherheitsziel abgedeckt ist.

78 Die Annahmen und die Ziele an die Umgebung entsprechen sich inhaltlich eindeutig, dass hierdurch sichergestellt ist, dass alle Annahmen durch Si-

cherheitsziele vollständig abgedeckt sind und umgekehrt. Durch die Annahmen A.1, A.2 und A.8 bzw. durch die Sicherheitsziele OE.1, OE.2 und OE.8 wird sichergestellt, dass der TOE vor unbefugtem Zugriff geschützt ist.

79 A.3, A.4 und A.8 bzw. OE.3, OE.4 und OE.8 sorgen dafür, dass bei der Installation weder ein verändertes System noch ein veränderter TOE eingesetzt werden.

80 A.3, A.4, A.7 und A.8 bzw. OE.3, OE.4, OE.7 und OE.8 stellt die Integrität des Systems und des TOE während der Installation und des Betriebs sicher.

81 A.5 und A.6 bzw. OE.5 und OE.6 stellen sicher, dass die TOE-Umgebung die benötigten Funktionen, Seed-Generierung und Hashfunktion, zur Verfügung stellt.

82 Das Ziel O.1 ist hinreichend für die Bedrohung T.1, da Zufallszahlen der Funktionalitätsklasse K3 die Eigenschaft haben, nicht vorhersagbar zu sein.

8.2 Erklärung der Sicherheitsanforderungen

8.2.1 Erklärung der funktionalen Sicherheitsanforderung

Erklärung der funktionalen Sicherheitsanforderung des TOE

83 Die funktionale Sicherheitsanforderung FCS_RND.1 ist geeignet, das Sicherheitsziel O.1 zu erfüllen, da die Qualitätsmetrik, die in FCS_RND.1.1 spezifiziert wird, exakt den Sicherheitszielen entspricht.

Tabelle 8: Erklärung der funktionalen Sicherheitsanforderungen

O.1	Der TOE muss Zufallszahlen erzeugen, die der Funktionalitätsklasse K3 [AIS 20] mit Mechanismenstärke „hoch“ genügen.	
	FCS_RND.1.1	Die TSF müssen einen Mechanismus bereitstellen, um Zufallszahlen zu generieren, die **der Funktionalitätsklasse K3 und Mechanismenstärke hoch, wie in [AIS 20] beschrieben**[13] entsprechen.

84 Der Gebrauch der generierten Zufallszahlen ist nicht Bestandteil weiterer Sicherheitsfunktionen des TOE, so dass die Komponente FCS_RND.1.2 keine weitere Bedeutung hat.

Erklärung der funktionalen Sicherheitsanforderung der IT-Umgebung des TOE

[13] Operation: Zuweisung „definierte Qualitätsmetrik“.

OE.5	Der TOE muss mit einem seed von ausreichend hoher Entropie ($H(p_A) \geq 80$) initialisiert werden.	
	FCS_RND.1.1	Die TSF müssen einen Mechanismus bereitstellen, um Zufallszahlen zu generieren, die **mindestens einer Entropie von 80**[14] entsprechen

85 Der Gebrauch der generierten Zufallszahlen ist nicht Bestandteil weiterer Sicherheitsfunktionen des TOE, so dass die Komponente FCS_RND.1.2 keine weitere Bedeutung hat. Die Anforderungen FPT_TST.1 und FPT_AMT.1 sind durch Abhängigkeiten hinzugekommen und stehen daher nur indirekt mit OE.5 im Zusammenhang.

OE.6	Es wird angenommen, dass die TOE-Umgebung den Hash-Algorithmus SHA-512 bereitstellt.	
	FCS_COP.1.1	The TSF shall perform **the computation of a hash-digest**[15] in accordance with a specified cryptographic algorithm **SHA-512**[16] and cryptographic key sizes **none** that meet the following: **Secure Hash Standard [NIST 180-2]**[17].

86 Der Gebrauch der generierten Zufallszahlen ist nicht Bestandteil weiterer Sicherheitsfunktionen des TOE, so dass die Komponente FCS_RND.1.2 keine weitere Bedeutung hat. Die Anforderungen FPT_TST.1 und _AMT.1 sind durch Abhängigkeiten hinzugekommen und stehen daher nur indirekt mit OE.5 im Zusammenhang.

87 Die Ziele OE.1, OE.2, OE.3, OE.4 und OE.8 werden durch Anforderungen an die nicht-IT-Umgebung des TOE erfüllt.

8.2.2 Analyse des Zusammenwirkens

Analyse des Zusammenwirkens (TOE)

88 Es gibt nur eine funktionale Anforderung, daher ist das Zusammenwirken formal erfüllt.

[14] Operation: Zuweisung „definierte Qualitätsmetrik“.

[15] Operation: assignment: list of cryptographic operations

[16] Operation: assignment: cryptographic algorithm

[17] Operation: assignment: list of standards

Analyse des Zusammenwirkens (IT-Umgebung)

89 Für die Ziele OE.5, OE.6 und OE.8 gibt es jeweils genau eine funktionale Anforderung. Daher ist jeweils das Zusammenwirken formal erfüllt.

8.2.3 Erklärung der Abhängigkeiten

Erklärung der Abhängigkeiten (TOE)

90 In FCS_RND.1 ist folgende Abhängigkeit beschrieben:

91 FPT_TST.1 TSF testing

92 Hierarchical to: No other components.

93 Dependencies: FPT_AMT.1 Abstract machine testing

94 FPT_TST.1.1 The TSF shall run a suite of self tests [selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]] to demonstrate the correct operation of the TSF. operation of [selection: [assignment: parts of TSF], the TSF].

95 FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of [selection: [assignment: parts of TSF], TSF data].

96 FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

97 Diese Abhängigkeit ist nur für physikalische Zufallszahlengeneratoren sinnvoll, da bei diesen während des Betriebs die Ausgabe überprüft werden muss, da die physikalische Quelle etwa durch äußere Einflüsse gestört werden könnte. Bei einem deterministischen Zufallszahlengenerator verändert sich die Qualität der erzeugten Zahlen nicht im Laufe der Zeit, sondern die Qualität der Zahlen hängt ausschließlich von der Güte des Algorithmus und des Seed ab. Diese beiden Faktoren werden bereits im Design festgelegt, so dass ein Selbsttest des TOE während des Betriebs überflüssig ist.

98 Die Integrität des Codes kann der vertrauenswürdige Administrator jederzeit über einen Prüfsummenabgleich durchführen, so dass hierzu keine Funktionalität des TOE bereitgestellt werden muss.

Erklärung der Abhängigkeiten (IT-Umgebung)

99 In FCS_RND.1 ist folgende Abhängigkeit beschrieben:

100 FPT_TST.1 TSF testing

101 Hierarchical to: No other components.

102 Dependencies: FPT_AMT.1 Abstract machine testing

103 FPT_TST.1.1 The TSF shall run a suite of self tests **periodically during normal operation**[18] to demonstrate the correct operation of **the TSF**[19].

104 FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of **the TSF**[20].

105 FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

106 Dependencies: FPT_AMT.1 Abstract machine testing

107 Bei der Seed-Generierung wird ein True Random Number Generator (TRNG) verwendet, so dass regelmäßig ein Selbsttest durchgeführt werden muss, der sicherstellt, dass es sich bei den erzeugten Zahlen tatsächlich um Zufallszahlen handelt. Hierfür sind statistische Tests geeignet. Die Entropie ist aus dem Design des TRNG abzuleiten. Aus dem Selbsttest leitet sich weiter die folgende Abhängigkeit ab:

108 FPT_AMT.1 Abstract machine testing

109 Hierarchical to: No other components.

110 FPT_AMT.1.1 The TSF shall run a suite of tests **periodically during normal operation**[21] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

111 Dependencies: No dependencies

112 Die Bereitstellung der Hashfunktion lässt sich in den CommonCriteria, Teil 2 [CC-Teil2] definieren mit der Familie FCS_COP:

113 FCS_COP.1 Cryptographic operation

114 Hierarchical to: No other components.

115 FCS_COP.1.1 The TSF shall perform **the computation of a hash-digest**[22] in accordance with a specified cryptographic algorithm **SHA-512**[23] and cryptographic key sizes **none** that meet the following: **Secure Hash Standard [NIST 180-2]**[24].

116 Dependencies: [FDP_ITC.1 Import of user data without security attributes

or

[18] Operation: selection

[19] Operation: selection

[20] Operation: selection

[21] Operation: selection

[22] Operation: assignment: list of cryptographic operations

[23] Operation: assignment: cryptographic algorithm

[24] Operation: assignment: list of standards

FDP_ITC.2 Import of user data with security attributes

or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

117 Die Verwendung des Hashalgorithmus steht hier nicht im Zusammenhang mit einer Kryptographischen Anwendung. Es wird kein kryptographischer Schlüssel generiert, so dass auch keiner zerstört werden müsste, es werden keine Sicherheitsattribute benötigt und es werden keine Benutzerdaten importiert. Daher sind die Abhängigkeiten FCS_CKM.1, FCS_CKM.4, FMT_MSA.2, FDP_ITC.1 und FDP_ITC.2 in dieser IT-Umgebung nicht relevant.

8.2.4 Erklärung der Mindest-Stärkestufe

118 Die vom TOE erzeugten Zufallszahlen dürfen nicht vorhersagbar sein. Nach [AIS20] ist das für deterministische RNGs erst ab Klasse K3 erreicht, und wird durch die hohe Mechanismenstärke unterstützt. Da es sich bei dem TOE um einen deterministischen RNG handelt, ist die Qualität des TOE stark abhängig von der Güte der Initialisierung. Das geeignete Maß für diese Güte ist die Entropie des Startwertes. Nach [AIS 20] gilt für eine Entropie größer als 80, dass die Mechanismenstärke „hoch“ für die Funktionalitätsklasse K3 erfüllt ist.

119 Aufgrund der Annahmen an die Umgebung sind mögliche Schwachstellen durch die externen Schnittstellen ausgeschlossen. Es ist deshalb trotz der hohen Mechanismenstärke ausreichend, eine Schwachstellenanalyse gemäß AVA_VLA.1 durchzuführen.

8.2.5 Erklärung zur Widerspruchsfreiheit und gegenseitigen Unterstützung

120 Die Wahl der IT-Sicherheitsanforderungen ist widerspruchsfrei, da es nur eine gibt.

8.2.6 Erklärung zu den Anforderungen an die Vertrauenswürdigkeit

121 Hinsichtlich Teil 3 der CC soll der Zufallszahlengenerator die Vertrauenswürdigkeitsstufe EAL 3+ erreichen. Zusätzlich zu EAL 3 ist aus Sicht des BSI eine Quelltextanalyse (ADV_IMP.1) notwendig. Des Weiteren ist es aufgrund von Abhängigkeiten innerhalb der CC erforderlich, bei EAL 3+ einen beschreibenden Entwurf auf niedriger Ebene (ADV_LLD.1) bereitzustellen und eine Dokumentation der Entwicklungswerkzeuge (ALC_TAT.1) zu erstellen.

8.3 Erklärung der TOE-Übersichtsspezifikation

8.3.1 Erfüllung der funktionalen Sicherheitsanforderungen

122 Die Sicherheitsfunktion SF1 erfüllt die Sicherheitsanforderungen, da die Eigenschaften von SF1, nämlich K3 und Mechanismenstärke „hoch“, genau den in FCS_RND.1.1 festgelegten Qualitätsmetriken entsprechen.

Tabelle 9: Erfüllung der funkt. Sicherheitsanforderungen

FCS_RND.1	Qualitätsmetrik für Zufallszahlen	
	SF1	Generierung von Zufallszahlen der Funktionalitätsklasse K3 gemäß [AIS 20] mit Mechanismenstärke „hoch“ incl. einer transitionfunction, die eine Zufallszahl aus dem Bereich [0, 2^{31}-2] liefert.

8.3.2 Konsistenz der Mechanismenstärke-Postulate

123 Um einen möglichst hohen Grad an Vertrauenswürdigkeit zu erreichen, müssen die Anforderungen an die Sicherheitsmaßnahmen und an die Maßnahmen zur Vertrauenswürdigkeit einander entsprechen. Das wird gewährleistet, indem zum einen an die Sicherheitsmaßnahmen die Forderung nach Sicherheitsklasse K3 mit hoher Mechanismenstärke gestellt wird, zum anderen die Anforderungen an EAL3 noch um die Familien ADV_IMP.1, ADV_LLD.1, und ALC_TAT.1 ergänzt werden.

8.3.3 Analyse des Zusammenwirkens der Sicherheitsfunktionen

124 Es gibt nur eine fkt. Anforderung, daher ist das Zusammenwirken formal erfüllt.

8.3.4 Erklärung zu den Maßnahmen der Vertrauenswürdigkeit

125 Der TOE erfüllt die Vertrauenswürdigkeitsanforderungen für die Evaluationsstufe EAL3+. Der Hersteller liefert im Rahmen der Evaluierung neben dem TOE (gemäß ATE_IND.1) folgende zusätzliche Dokumente, um eindeutig die Erfüllung der Anforderungen entsprechend EAL3+ nachzuweisen.

- Dokumentation Auslieferung und Betrieb (gemäß ADO_DEL.1 und ADO_IGS.1)
- Dokumentation Entwicklung (gemäß ADV_FSP.1, ADV_HLD.2, ADV_LLD.1, ADV_IMP.1, ADV_RCR.1)
- Dokumentation Handbücher (gemäß AGD_ADM.1, AGD_USR.1)
- Dokumentation Konfigurationsmanagement (gemäß ACM_CAP.3, ACM_SCP.1)
- Dokumentation Lebenszyklus-Unterstützung (gemäß ALC_DVS.1, ALC_TAT.1)
- Testdokumentation (gemäß ATE_COV.2, ATE_DPT.1, ATE_FUN.1, ATE_IND.2)
- Dokumentation Schwachstellenbewertung (gemäß AVA_MSU.1, AVA_SOF.1, AVA_VLA.1)

9 Anhang: Familie FCS_RND

126 Die Familie FCS_RND wird in „Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren“ [AIS 31], wie folgt definiert:

127 FCS_RND Erzeugung von Zufallszahlen

128 Familienverhalten

129 Diese Familie definiert Qualitätsmetriken für die Erzeugung von Zufallszahlen, die für kryptographische Zwecke vorgesehen sind.

130 Komponentenabstufung

131 FCS_RND.1 Generierung von Zufallszahlen durch TSF erfordert, daß die Zufallszahlen die definierte Qualitätsmetriken erfüllen.

132 **Management: FCS_RND.1**

133 Es sind keine Managementfunktionen vorgesehen.

134 **Protokollierung: FCS_RND.1**

135 Es sind keine Aktionen definiert, die prokolliert werden sollen.

136 **FCS_RND.1** Qualitätsmetrik für Zufallszahlen

137 Ist hierarchisch zu: Keinen anderen Komponenten.

138 FCS_RND.1.1 Die TSF müssen einen Mechanismus bereitstellen, um Zufallszahlen zu generieren, die [Zuweisung: definierte Qualitätsmetrik] entsprechen.

139 FCS_RND.1.2 Die TSF müssen in der Lage sein, den Gebrauch der TSF-generierten Zufallszahlen für [Zuweisung: Liste der TSF-Funktionen] durchzusetzen.

140 Abhängigkeiten: FPT_TST.1 TSF testing.

10 Glossar

141	A.*	Assumption
142	CC	Common Criteria
143	EAL	Evaluation Assurance Level
144	O.*	Objective
145	OE.*	Objective on the (TOE-)Environment
146	OS	Operating System
147	PP	Protection Profile
148	SF	Security Function
149	ST	Security Target
150	T.*	Threat

151 TOE Target of Evaluation

Literatur

[AIS 20] Bundesamt für Sicherheit in der Informationstechnik (BSI), „Anwendungen und Interpretationen zum Schema (AIS): Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren“, AIS 20, Version 1, 02.12.1999.

[AIS 31] Bundesamt für Sicherheit in der Informationstechnik (BSI), „Anwendungen und Interpretationen zum Schema (AIS): Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren“, AIS 31, Version 1, 25.09.2001.

[BNA 2006] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen, „Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen), 2.Januar 2006, veröffentlicht im Bundesanzeiger Nr.58, S. 1913-1915, 23.03.2006.

[BSI PP0002] Atmel Smart Card ICs, Hitachi Europe Ltd. , Infineon Technologies AG and Philips Semiconductors, “Smartcard IC Platform Protection Profile”, version 1.0, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0002, Juli 2001.

[CC-Teil1] Common Criteria, „Common Criteria for Information technology security evaluation, Part 1: Introduction and general model“, Version 2.3, August 2005.

[CC-Teil2] Common Criteria, „Common Criteria for Information technology security evaluation, Part 2: Security functional requirements“, Version 2.3, August 2005.

[CC-Teil3] Common Criteria, „Common Criteria for Information technology security evaluation, Part 3: Security assurance requirements“, Version 2.3, August 2005.

[DIEHARD] George Marsaglia,”The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness”, http://www.stat.fsu.edu/pub/diehard/, 1995.

[IACR 86] Zvi Gutterman, Benny Pinkas and Tzachy Reinman, “Analysis of the Linux Random Number Generator“, Cryptology ePrint Archive (http://eprint.iacr.org/), Report 2006/086, May 2006.

[MT19937] Matsumoto, Makoto and Takuji Nishimura, „Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator”, ACM Trans. Model. Comput. Simul. 8, No.1, pp. 3-30 (1998).

[NIST 180-2] NIST, „FIPS Publication 180-2: Secure Hash Standard (SHS)“, August 2002 und change Notice 1, Februar 2004.

[RNG-Develop] Casinoland GmbH, “Der Zufallszahlengenerator des Online-Casinos der Casinoland GmbH - Entwicklung (ADV) / Konzept”, Version 1.3, 28.11.2006.

[Syste] Casinoland GmbH, „Systemverwalterhandbuch Casinoland-RNG (Zufallszahlengenerator RNG.java)“, Version 1.7, 19.12.2007

[TGSFR] M. Matsumoto and Y. Kurita, "Twisted GFSR generators II", ACM Trans. on Modeling and Computer Simulation, 4(1994),pp. 254-266, 1994.