Professionelle Zufallsgeneratoren für kryptografisch sichere Zufallszahlen

Zusammenfassung

Kryptografische Verfahren benötigen prinzipiell zwei Komponenten: den (meist) veröffentlichten Verschlüsselungsalgorithmus und den geheimen Schlüssel. In IT-Sicherheitsapplikationen werden die geheimen Schlüssel mittels Zufallsgeneratoren erzeugt. Die Schlüssel dürfen unter keinen Umständen vorhersagbar oder rekonstruierbar sein. Schwächen verwendeter Zufallsgeneratoren, vor allem in professionellen IT-Sicherheitsapplikationen, sind in Publikationen vielfältig illustriert. In Deutschland hat die Regulierungsbehörde für IT-Sicherheit (Bundesnetzagentur BNetzA) verbindliche Vorgaben über die Verwendung von Zufallsgeneratoren der Funktionalitätsklassen PTG.3 und DRG.4 in IT-Sicherheitsapplikationen festgelegt.
Im Folgenden werden hybride Zufallsgeneratoren der Klasse PTG.3 des Autors vorgestellt, die als einzige, der auf dem kommerziellen Markt verfügbaren Zufallsgeneratoren, diese Anforderungen erfüllen.

IBB Ing.-Büro Bergmann
Sonnenweg 3
D-15537 Grünheide

Mobil: 0172 308 6554
Internet: www.ibbergmann.org
eMail: info@ibbergmann.org

1. Stand der Forschung und Technik

Zufallszahlen sind das Fundament vieler kryptographischer Verfahren und Protokolle. Es ist wichtig, dass die verwendeten Zufallszahlen nicht vorhersagbar sind. Solche Zufallszahlen zu erzeugen, fällt Computern naturgemäß schwer. Zahlreiche Meldungen kritisieren Lücken, Schwächen und Manipulationen bei der Erzeugung von Zufallszahlen für kryptografische Verfahren. Hier einige typische Beispiele:

- RSA Security warnt vor NSA-Zufallsgenerator (1)
- Die Herkunft der NIST-Kurven (2)
- NIST rät von Dual_EC_DRBG wegen möglicher NSA-Backdoor ab (3)
- NIST lässt Zufalls-Generatoren neu prüfen (4)
- NetBSD erzeugt schwache Schlüssel (5)
- PHP stümpert bei Zufallszahlen (6)
- Mathematiker entlarvt schwache DKIM-Schlüssel (7)
- MIPS-Router mit Entropieproblemen (8)
- OpenSSL erzeugt zu oft den gleichen Zufall (9)
- Verschlüsselungsstandard unter Backdoor-Verdacht (10)
- Zu wenig Zufall im Zufallszahlengenerator von OpenBSD (11)
- Apache-Tool erzeugt Passwort-Hashes mit vorhersagbaren Salts (12)
- Gute Zahlen, schlechte Zahlen (13)
- NSA-Affäre: Generatoren für Zufallszahlen unter der Lupe (14)
- Misstrauen bei RNGs von Intel und VIA (15)
- JavaScript-Engine V8: Vorsicht vor Math.random() (16)

Die Aufzählung ist nicht vollständig. Bei allen Meldungen geht es nicht um spezielle, bedeutungslose Applikationen, sondern um millionenfach installierte Standardprogramme in professionellen Anwendungen. Vor allem dort, wo kontinuierlich viele Zufallszahlen benötigt werden (Netzwerke, Kommunikationssysteme), sind Angriffe auf schwache Zufallsgeneratoren am erfolgreichsten.

Nach dem Kerckhoff-Prinzip (die Sicherheit soll nur auf der Geheimhaltung des Schlüssels beruhen, nicht auf der Geheimhaltung des kryptographischen Algorithmus) benötigt jede Art von Verschlüsselung eine geheime Komponente, die *unter keinen Umständen* vorhersagbar oder rekonstruierbar sein darf: der aus Zufallszahlen gebildete Schlüssel.

In der Praxis werden für die Erzeugung von Zufallszahlen drei grundsätzliche Varianten genutzt:

- Pseudozufallsgeneratoren (DRNG, DRG)
- Physikalische Zufallsgeneratoren (TRNG)
- Hybride Zufallsgeneratoren (PTRNG, PTG)

Pseudozufallsgeneratoren

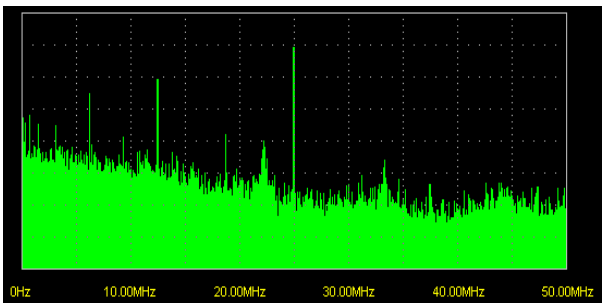
Quelle der Generierung von Pseudozufall ist ein Startwert (z.B. bestehend aus Passwort, Timer-Register, Tastaturanschlägen, Mausbewegungen usw.), mit dem ein mathematisch-kryptografischer Algorithmus eine statistisch gut verteilte Zufallsfolge erzeugt. Die gesamte Sicherheit der per

Pseudozufall erzeugten geheimen Schlüssel hängt *ausschließlich* von dieser Anfangsinitialisierung ab. Die Anfangsinitialisierung ist bei richtiger Wahl der Quelle der einzige wirklich zufällige Parameter. Alles Weitere ist *deterministisch* und somit berechenbar. Eine schwache Anfangsinitialisierung (ein trivialer Startwert, Seed) ist im statistischen Ergebnis nicht erkennbar, aber ein effizienter Angriffspunkt der Kryptoanalysis.

Auch professionelle Entwickler nutzen als Seed für Pseudozufall oftmals das Timer-Register in der Annahme: wer will denn schon wissen, in welcher Sekunde das Register ausgelesen wurde. Für einen Angreifer kein Problem, denn ein Jahr hat ca. 32 Millionen Sekunden. Um diese mit der totalen Probiermethode (brute force) durchzutesten, benötigt man nur eine durchschnittliche Rechenleistung. Wird der gleiche Seed mehrfach verwendet (siehe Meldung vom 18.02.2008: http://www.heise.de/newsticker/meldung/Zu-wenig-Zufall-im-Zufallszahlengenerator-von-OpenBSD-178124.html (11)), so entstehen schlüsselgleiche Geheimtexte. Ein sicherer Erfolg für die Kryptoanalysis.

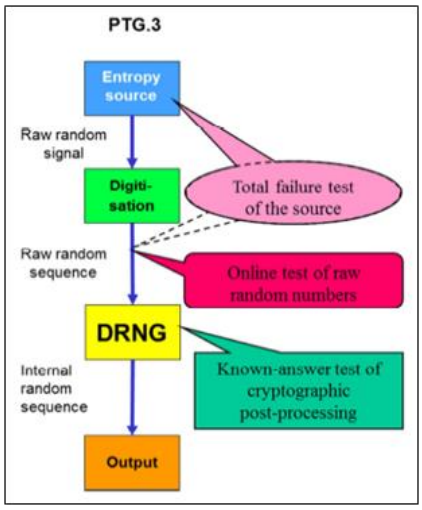
Physikalische Zufallsgeneratoren

Bekannte Lösungen für die Generierung physikalischen Zufalls arbeiten meist auf Basis thermischer Rauschquellen (Widerstände, Z-Dioden, Transistoren), freilaufender Oszillatoren die mit dem Systemtakt korrelieren oder quantenoptischer Quellen. Gemeinsames Problem aller bisher bekannten Lösungen sind die den Rauschquellen überlagerten deterministischen Takte (Systemtakte, Schalttakte logischer Schaltkreise, Netzfrequenz usw.).



Im Bild rechts ist die Spektralanalyse (FFT) einer verstärkten Rauschquelle mit überlagerten Gleichtakten dargestellt. Die Rauschquelle im physikalische Zufallsgenerator R300A der schwedischen Firma Protego (http://www.protego.se) beispielsweise wird ca. 15.000-fach verstärkt, um einen nutzbaren Pegel zu erreichen. Damit werden Störsignale der Stromversorgung automatisch mit verstärkt. Derart genutzte Zufallsquellen haben dadurch eine zu geringe Entropie und unzureichende Gleichverteilung der Bits und Bytes und erfüllen nicht die von deutschen Kryptologen geforderten AIS31-Normen (BSI: Interpretationen AIS31 (18)) für unbearbeitete Zufallsdaten (Rohdaten).

Hybride Zufallszahlengeneratoren



Hybride Zufallszahlengeneratoren vereinen Sicherheits-eigenschaften von deterministischen und physikalischen Zufallszahlengeneratoren und besitzen neben einer starken Rauschquelle eine starke kryptographische Nachbearbeitung mit Gedächtnis. Die Klasse PTG.3 stellt nach Algorithmenkatalog 2015 (19) der Bundesnetzagentur (BNetzA) die *stärkste Funktionalitätsklasse* dar.

Für die Erzeugung und Bewertung von kryptografisch sicheren Zufallszahlen sind eindeutige Normen und Regeln vorgegeben. Diese sind neben dem Algorithmenkatalog in den AIS31-Dokumenten des Bundesamtes für Sicherheit in der Informationstechnik (BSI) festgelegt und bestimmen die notwendigen Eigenschaften von

unbearbeiteten Zufallszahlen (Rohdaten): „*Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren*“ (21). Demnach muss ein kryptografisch sicherer Zufallsgenerator z.B. eine Entropie von mehr als 7,97 Bit/Byte haben. Weiterhin müssen die Zufallsbits bestimmten Eigenschaften genügen, wie die Gleichverteilung, Bitunabhängigkeit und die Bewertung von 2-, 3-, und 4-Bit Zufallsfolgen.

Grundsätzlich verlangt die Bundesnetzagentur im aktuellen Algorithmenkatalog:
„Für Zertifizierungsdienstanbieter wird die Verwendung von Zufallsgeneratoren der Funktionalitätsklassen *PTG.3* und DRG.4 im Grundsatz *ab 2015 verpflichtend* werden, sowohl allgemein bei der Erzeugung von Langzeitschlüsseln als auch bei der Erzeugung von Ephemeralschlüsseln.“

2. AIS31-kompatible Lösung zur Klasse PTG.3 der Firma IBB

Grundlage der im Folgenden vorgestellten Lösungen zur Klasse PTG.3 ist ein *stochastisches Modell*, mit dem die Leistungsfähigkeit zur Generierung kryptografisch sicherer Zufallszahlen begründet wird. Dieses Modell erklärt:

- die robuste und hohe Entropie der Rauschquelle
- das Prinzip der Abtastung des analogen Rauschsignals
- die permanente Überwachung der Rauschquelle durch Frequenzmessung
- die kryptografische Nachbearbeitung durch Mayer-Einwegfunktionen (Schema)
- den permanenten statistischen Online-Test der Rohdaten

Durch die Firma IBB wurde eine patentierte (EU-Patent EP 150 98 38), stabile und reproduzierbare Lösung für die Generierung von echtem Zufall auf Basis physikalischer Rauschquellen mit hoher Entropie (>7,997 Bit/Byte) entwickelt. Keine der weltweit bekannten reproduzierbaren kommerziellen Applikationen erreicht diesen Wert. Die Lösung der Firma IBB besteht damit erstmalig die vom BSI vorgegebenen AIS31-Normen.
Entscheidende Änderungen zu bekannten Applikationen sind:

- Verwendung von zwei Z-Dioden als thermische Rauschquellen; damit werden Streuungen des nichtspezifizierten Parameters Rauschspannung kompensiert
- Einsatz von Z-Dioden höherer Durchbruchsspannung, da diese Typen technologisch bedingt eine größere Rauschspannung erzeugen
- Verstärkung der beiden Rauschsignale durch einen Differenzverstärker mit hoher Gleichtaktunterdrückung (sehr wirksame Störsignalunterdrückung)
- Arbeitspunktstabilisierung durch Signalrückkopplung
- das Rauschsignal wird nur ca. 300-fach verstärkt; damit unempfindlich gegen elektromagnetische Einstrahlung
- spezielle Schutzmaßnahmen wie Schirmung oder Filterung der Versorgungsspannung sind nicht notwendig

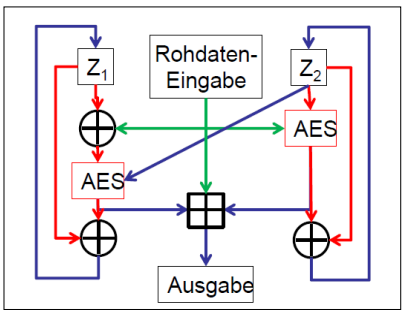
Folgende Eigenschaften werden reproduzierbar gesichert:

- sehr hohe Entropie der Rohdaten (>7,997 Bit/Byte)
- hohe Frequenz des digitalisierten Rauschsignals mit 7..8 Mhz
- Ausgabegeschwindigkeit der Rohdaten > 330 Kbit/s
- Ausgabegeschwindigkeit in der Klasse PTG.3 > 140 Kbit/s
- keine Bit-Abhängigkeiten der Rohdaten nachweisbar

- besteht alle Rohdaten-Tests nach AIS31 im Temperaturbereich von -60..+110°C
- kontinuierlicher automatischer Selbstabgleich aller Arbeitspunkte

Zur Evaluierung wurden umfangreiche statistische Tests durchgeführt. Bereits die statistischen Analysen der Rohdaten zeigten eine sehr hohe Entropie und keine nachweisbaren Abhängigkeiten der Zufallsbits. Die mit empfohlenen Methoden nach Schema zur Klasse PTG.3 (21) nachbearbeiteten Zufallsdaten wurden mit internationalen Referenztests, wie der Diehard-Test-Suite nach George Marsaglia, der NIST-Test-Suite und einem weiteren eigenen statistischen Basistest, umfangreich untersucht. Keine der generierten Testfolgen konnte Unterschiede zu den Ergebnissen eines idealen Zufallsgenerators aufzeigen.

Die kryptografische Nachbereitung der Rohdaten nach PTG.3 sind im Schema rechts dargestellt. Die Rohdaten-Eingabe erfolgt mit 16 Byte Rohdaten des physikalischen Zufallsgenerators und ergeben nach der Verarbeitung 16 Byte Ausgabedaten. Es wird also *explizit kein Pseudozufall* generiert. Einschätzungen zur Qualität geben statistische Analysen für Rohdaten Zufallsdaten auf der Homepage des Autors.



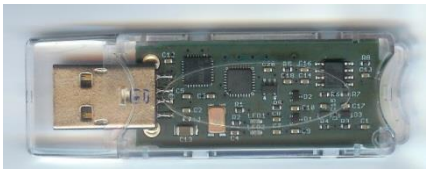
Alle aktuellen Lösungen des IBB entsprechen den Forderungen der AIS31-Normen und der Klasse PTG.3 aus dem Algorithmenkatalog 2015. Bestanteile der Funktionalität aller Applikationen sind *Sicherheitseigenschaften und Überwachungsfunktionen* die garantieren, dass der Anwender keine eigenen Kontrollen in seine Anwendungen implementieren muss. Werden Fehler im Zufallsgenerator detektiert, wird die Zufallsausgabe sofort eingestellt und eine Fehlermeldung generiert.

3. Verfügbare Zufallsgeneratoren der Firma IBB

Die folgenden Applikationen besitzen die oben beschriebenen Eigenschaften. Sie wurden umfangreich erprobt und statistisch analysiert.

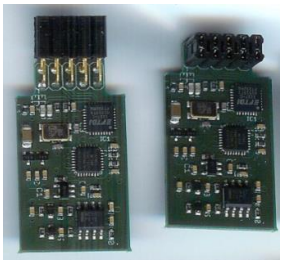
PRG310-4

- Der PRG310-4 ist für den *externen* Anschluss an beliebige PC-Systeme mit einem Standard-USB-Anschluss vorgesehen.



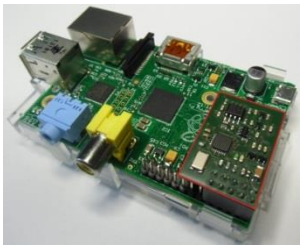
PRG310-5

- Der bis auf den Steckverbinder vollkommen schaltungs- und funktionsgleiche PRG310-5 ist für den Einsatz *innerhalb* beliebiger PC-Systeme konzipiert. Interne USB-Anschlüsse sind auf Mainbords als Pfostensteckverbinder ausgeführt. Die Pfostensteckverbinder auf dem PRG310-5-Modul sind als horizontale und vertikale Ausführung verfügbar.



PRG260

- Für den bekannten Raspberry-PI-Computer und kompatible wurde der PRG260 (rot umrandet) mit einem asynchronen Interface (UART) entwickelt, der funktionskompatibel zum PRG310 ist.

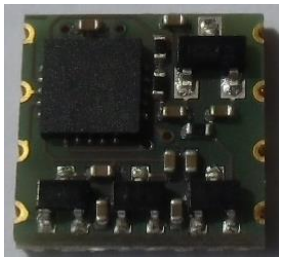


PRG600

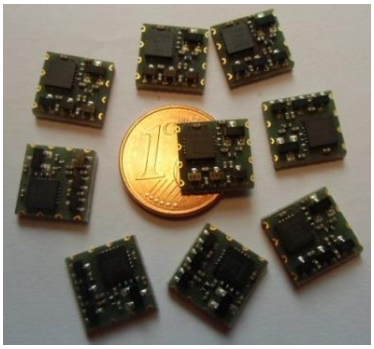
Aktuelle Untersuchungen auf dem Gebiet der Rauscherzeugung führten zu neuen Erkenntnissen. Empirisch ermittelte Halbleiter können reproduzierbar sehr hohe Rauschspannungen (>400mVss) bei einem breiten Rauschspektrum erzeugen, so dass keine Verstärkung erforderlich ist und der Signalpegel deutlich über dem Störpegel elektronischer Schaltungen liegt. Die Rauschquelle muss nicht ausgemessen werden, da sie, technologisch bedingt, immer gleiche Rauschamplituden und ein gleiches Frequenzspektrum erzeugt. Ursache des Rauschens ist ein ausgeprägter stabiler Avalanche-Effekt in einem definierten Arbeitspunkt.

Aus diesen Erkenntnissen wurde eine neue Applikation entwickelt, erprobt und analysiert, die alle erforderlichen Eigenschaften eines Zufallsgenerators der Klasse PTG.3 repräsentiert. Unter der Bezeichnung PRG600 steht ein neuer Zufallsgenerator mit folgenden Eigenschaften zur Anwendung bereit:

- Basis ist ein *stochastisches Modell* mit allen erforderlichen Sicherheitsfunktionen (statistischer Online-Test, Überwachung der Rauschquelle durch Frequenzmessung, Blockierung der Datenausgabe bei Fehlermeldung)
- Entropiequelle ist ein Standard-Halbleiterbauelement
- *Entropie >7,99* Bit/Byte, aus Rohdaten ermittelt
- 0/1-Verhältnis der Ausgabedaten *0,499..0,501* (> 100 KByte)
- Synchrones SPI-Interface mit einer Datenausgabegeschwindigkeit von 40 Kbit/s, umschaltbar auf 4 Kbit/s für den Einsatz in zeitkritischen Applikationen
- Stabil im Temperaturbereich *-20°C..+85°C*
- Abmessungen *10*10*2 (mm)*
- Stromaufnahme bei 3,3V: *<4mA, im Sleep-Modus <15µA*



In einer minimalen Anschlussbelegung werden neben der Stromversorgung nur das Zufallssignal und das synchrone Taktsignal benötigt, um permanent Zufallsdaten mit 40 Kbit/s auszugegeben. Wird ein Fehler (Ausfall der Rauschquelle, statistischer Online-Test fehlerhaft) erkannt, wird die Ausgabe der Zufallsdaten blockiert und ein Fehlersignal aktiviert.
Statistische Analysen für Rohdaten und Zufallsdaten stehen auf der Homepage des Autors zur Verfügung.



4. Anwendungen

Die PRG310-Varianten sind die leistungsfähigsten Applikationen mit den breitesten Anwendungsmöglichkeiten. Hier vereinen sich höchste Sicherheit auf Basis eines stochastischen Modells, höchste Entropie, alle Möglichkeiten der Zufallsgenerierung von Rohdaten bis Zufallsdaten der Klassen PTG.2 und PTG.3 sowie hohe Ausgabegeschwindigkeit. Es sind weltweit keine kommerziellen Applikationen bekannt, die auch nur annähernd diese Qualität und Möglichkeiten bieten. Damit sind die PRG310-Applikationen bestens für breite Anwendungen in der Zufallserzeugung geeignet.

Alle aufgeführten Zufallsgeneratoren sind in der Lage, höchste Anforderungen an ein One-Time-Pad-Verfahren (OTP) zu erfüllen. Hat der Schlüssel in seiner Statistik keine Schwächen und ist mit physikalischem Zufall generiert, kann das Verschlüsselungsverfahren bei bestimmungsmäßiger Handhabung prinzipiell nicht gebrochen werden. Mit diesem einzigen beweisbar sicheren Verschlüsselungsverfahren lassen sich beispielsweise sehr sichere Lösungen gegen Industrie- und Wirtschaftsspionage realisieren.

Anwendungen des PRG310 erfolgen bereits in verschiedenen Rechenzentren zur Datenverschlüsselung von Netzwerken. Die permanent erforderlichen Schlüssel werden durch den Zufallsgenerator mit Funktionen der Klasse PTG.3 erzeugt.

Weitere exemplarische Beispiele für die PRG310- und PRG260-Applikationen:

- Netzwerksicherheit
- Transaktionen beim Homebanking (SSL-Verschlüsselung)
- Internet-Verschlüsselung
- elektronischer Zahlungsverkehr
- Erstellung von PKI-Zertifikaten
- OneTimePad-Verfahren

Der PRG600 ist mit seinen Parametern bestens geeignet, Sicherheitsverfahren in mobilen Applikationen mit höchsten Sicherheitsanforderungen zu realisieren.

Der Einsatz des PRG600 ist optimal in folgenden Anwendungen

- Sichere Kommunikation in mobilen Geräten (Smartphones, Tablets)
- einfachere Administration und sichere Verschlüsselung bei drahtloser Datenübertragung: WLAN, Bluetooth, GSM, ZiggBee, Industriedatenfunk
- Einfache Implementierung in proprietären Applikationen

Generell wird für die Generierung von kryptografisch sicheren Zufallszahlen empfohlen, *keine* Lösung zu verwenden, die auf der Basis von Closed-Source-Hardware aus dem nichteuropäischen Ausland arbeitet oder auf den von der Computer Security Division am NIST der USA für den praktischen Einsatz empfohlenen Zufallserzeugungen beruht. Die am Anfang aufgeführten Beispiele fordern regelrecht derartige Festlegungen.

5. Verweise

(1) http://www.golem.de/news/bsafe-rsa-security-warnt-vor-nsa-zufallsgenerator-1309-101727.html
(2) http://www.golem.de/news/elliptische-kurven-die-herkunft-der-nist-kurven-1309-101567.html
(3) http://www.golem.de/news/verschluesselung-nist-raet-von-dual-ec-drbg-wegen-moeglicher-nsa-backdoor-ab-1309-101521.html
(4) http://www.heise.de/newsticker/meldung/NIST-laesst-Zufalls-Generatoren-neu-pruefen-1954677.html
(5) http://www.golem.de/news/fehler-im-zufallsgenerator-netbsd-erzeugt-schwache-schluessel-1303-98350.html
(6) http://www.heise.de/newsticker/meldung/PHP-stuempert-bei-Zufallszahlen-967062.html
(7) http://www.heise.de/security/meldung/Mathematiker-entlarvt-schwache-DKIM-Schluessel-1736107.html
(8) http://www.heise.de/security/meldung/MIPS-Router-mit-Entropieproblemen-1953097.html
(9) http://www.heise.de/security/meldung/OpenSSL-erzeugt-zu-oft-den-gleichen-Zufall-1942299.html
(10) http://www.heise.de/newsticker/meldung/Verschluesselungsstandard-unter-Backdoor-Verdacht-196659.html
(11) http://www.heise.de/newsticker/meldung/Zu-wenig-Zufall-im-Zufallszahlengenerator-von-OpenBSD-178124.html
(12) http://www.heise.de/newsticker/meldung/Apache-Tool-erzeugt-Passwort-Hashes-mit-vorhersagbaren-Salts-180864.html
(13) http://www.heise.de/security/artikel/Gute-Zahlen-schlechte-Zahlen-270078.html
(14) http://www.heise.de/security/meldung/NSA-Affaere-Generatoren-fuer-Zufallszahlen-unter-der-Lupe-1953716.html
(15) http://www.golem.de/news/freebsd-misstrauen-bei-rngs-von-intel-und-via-1312-103305.html
(16) http://www.heise.de/newsticker/meldung/JavaScript-Engine-V8-Vorsicht-vor-Math-random-3010353.html
(17) http://www.golem.de/news/verschluesselung-2013-das-jahr-der-kryptokalypse-1312-103617-2.html
(18) https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_pdf.html
(19) (http://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/QES/Veroeffentlichungen/Algorithmen/2015Algorithmenkatalog.pdf;jsessionid=2EFC232C278CA15479747CF5FB36D32C?__blob=publicationFile&v=1
(20) Dichtl, M.: How to predict the output of a hardware random number generator. In: Walter, D.C., Ko»c, C» .K., Paar, C. (eds.) CHES 2003. LNCS, vol. 2779, pp. 181{188.
(21) https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_31_Functionality_classes_for_random_number_generators_e.pdf?__blob=publicationFile