Empfehlung „Geeignete Kryptoalgorithmen" gemäß §17 (1) SigG

Stellungnahme

Dirk Fox, Stefan Kelm, Hans-Joachim Knobloch, Dr. Markus Michels, Dr. Holger Petersen
Secorvo Security Consulting GmbH

Version 1.0
Stand 24. Oktober 2002

Inhaltsübersicht

Abkürzungen

ANSI	American National Standards Institute
CEN	European Committee for Standardization
DRNG	Deterministic RNG
DSA	Digital Signature Standard
ETSI	European Telecommunications Standards Institute
FIPS	Federal Information Processing Standards
GNFS	General Number Field Sieve
ISO	International Organization for Standardization
MIPS	Millions of Instructions Per Second
MJ	MIPS-Jahre
NIST	National Institute of Standards and Technology
PDA	Personal Digital Assistant
RNG	Random Number Generator
RSA	Asymmetrisches Kryptosystem von Rivest, Shamir und Adleman, 1978
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SigG	Deutsches Signaturgesetz vom 22.05.2001
SigV	Signaturverordnung zum Deutschen Signaturgesetz vom 22.11.2001

1 Zusammenfassung

Der folgende Text nimmt Stellung zum Entwurf vom 09.09.2002 der Empfehlungen geeigneter Kryptoalgorithmen gemäß § 17 (1) des Signaturgesetztes (SigG) vom 22. Mai 2001 in Verbindung mit Anlage 1, I 2, Signaturverordnung (SigV) vom 22. November 2001, der zur Veröffentlichung im Bundesanzeiger vorgesehen ist.

Ein großer Teil der im Folgenden geäußerten Kommentare zur Algorithmenempfehlung findet sich bereits in einer Stellungnahme der Autoren zu der Empfehlung des BSI vom 24.01.2002 [FKMP_02]. Da diese Kommentare im aktuellen Empfehlungsentwurf überwiegend nicht berücksichtigt wurden, wir diese Punkte jedoch für besonders wichtig halten, werden sie an dieser Stelle erneut wiedergegeben.

2 Kommentierung des Empfehlungsentwurfs vom 09.09.2002

Bei der Durchsicht des Empfehlungsentwurfs fällt zunächst auf, dass das jeweils in den Einzelempfehlungen vorgeschlagene Sicherheitsniveau nicht durchgängig ist. Getreu dem Prinzip, dass der erreichbare Schutz immer von der Stärke der schwächsten Komponente bestimmt wird, zielt die folgende Kommentierung daher insbesondere auf die Formulierung eines einheitlichen Sicherheitsniveaus für alle kryptografischen Komponenten.

2.1 Vorwort

Der Hinweis auf die potenzielle Veröffentlichung eines Algorithmendokuments im Amtsblatt der Europäischen Gemeinschaften ist an dieser Stelle in zweifacher Hinsicht irreführend und sollte daher gestrichen werden:

- Bei dem Entwurf, welcher derzeit der EU-Kommission vorliegt, handelt es sich um ein Dokument, welches in einigen Punkten inhaltlich von der BSI-Empfehlung abweicht. Die beiden Dokumente stehen (obwohl z.T. dieselben Autoren daran mitgewirkt haben) in keiner direkten Beziehung, da in das EU-Dokument insbesondere auch die Anforderungen anderer europäischer Mitgliedsstaaten einfließen.
- Der formelle Status des der Kommission vorliegenden Algorithmendokuments ist derzeit unklar. Da dieses Dokument nicht innerhalb einer offiziellen Arbeitsgruppe von ETSI oder CEN erarbeitet wurde, ist eine Veröffentlichung im EU-Amtsblatt sehr unwahrscheinlich.[1]

2.2 Kryptographische Anforderungen (Abschnitt 1)

Keine Kommentare.

2.3 Vorschläge für geeignete Hashfunktionen (Abschnitt 2)

Die Sicherheit von Hashfunktionen mit 160 bit Ausgabewert betrachten wir als hinreichend, da 2n-bit Hashfunktionen nach dem Geburtstagsparadoxon eine Kollisionsresistenz von etwa 2^{-n} gegen Brute-Force-Angriffe bieten. Damit entsprechen 160-bit Hashfunktionen hinsichtlich

1 Vgl. das Protokoll eines Treffens des sog. „Artikel-9-Ausschusses“ [A9C_02].

des Sicherheitsparameters Blockchiffren mit 80-Bit Schlüssellänge, die gemäß der Lenstra-Verheul Studie [LeVe_99] als hinreichend sicher bis zum Jahr 2012 gelten.

Soll eine höhere Sicherheit erreicht werden, so müssten beispielsweise beim DSA und bei DSA-Varianten (vgl. Abschnitt 3.2 und 3.3 der Algorithmenempfehlung) die Mindestlänge des Schlüsselparameters *q* von 160 auf 180 bit erhöht werden. Damit nicht die Hashfunktion zum schwächsten Glied der Kette wird, müsste die Länge des Ausgabewertes der Hashfunktion ebenfalls auf 180 bit angehoben werden.

Die Hashfunktionen RIPEMD-160 und SHA-1 würden damit für eine weitere Nutzung ausscheiden; sie ließen sich derzeit nur durch die Anfang August 2002 als NIST-Standard verabschiedete Hashfunktion SHA-256 ersetzen [NIST_02]. Weitere aussichtsreiche, allerdings bislang nicht standardisierte Kandidaten für Hashfunktionen mit Ausgabewerten von mehr als 160 bit Länge sind derzeit:

- Die 512-bit Hashfunktion Whirlpool [BaRi_00], die im Rahmen des von der Europäischen Kommission geförderten Projekts NESSIE untersucht wird.
- Die MDC-2 und MDC-4 Schemata zur Konstruktion von 2n-Bit Hashfunktionen auf der Grundlage von n-bit Blockchiffren, die in Verbindung mit dem AES eine 256-bit Hashfunktion ergeben.

2.4 Vorschläge für geeignete Signaturalgorithmen (Abschnitt 3)

2.4.1 RSA (Abschnitt 3.1)

Die für das RSA-Verfahren in der tabellarischen Übersicht geforderten und empfohlenen Schlüssellängen erscheinen nicht schlüssig und decken sich auch nicht mit den Empfehlungen unterschiedlicher Experten und Standards.

- Grundsätzlich erscheint zunächst die Annahme vernünftig, dass die Faktorisierungserfolge in den kommenden Jahren nicht die Entwicklungen der vergangenen zwanzig Jahre übertreffen werden. Dafür spricht vor allem, dass seit der Entwicklung des Zahlkörpersiebs (Number Field Sieve) vor mehr als 10 Jahren durch J. M. Pollard (1990) wurden keine prinzipiell neuen Ansätze zur Faktorisierung großer Zahlen mehr publiziert wurden. Alle seitdem veröffentlichten Vorschläge betrafen Detailverbesserungen des Algorithmus. Mit dem "General Number Field Sieve" (GNFS) fand die theoretische Weiterentwicklung der Faktorisierungsalgorithmen 1993 einen bis heute gültigen Abschluss [BuLP_93].

Zukünftige Faktorisierungserfolge sind daher realistischer Weise vor allem auf Grund einer allgemeinen Zunahme der Leistungsfähigkeit von Rechnersystemen sowie der Möglichkeit zur Nutzung größerer, über das Internet verbundener Rechen-Cluster zu erwarten.

Grundsätzlich kann angenommen werden, dass die für einen verdeckten Angriff zur Verfügung stehende Rechenleistung deutlich unter der liegt, die für eine Faktorisierung mit Bündelung verteilter, freiwillig bereitgestellter Ressourcen im Internet zur Verfügung steht. Daher kann die im folgenden Bild dargestellte Abschätzung der Entwicklung der verfügbaren Rechenleistung für "öffentliche" Faktorisierungsangriffe als Obergrenze der bei einem Angreifer zu erwartenden Leistung betrachtet werden.

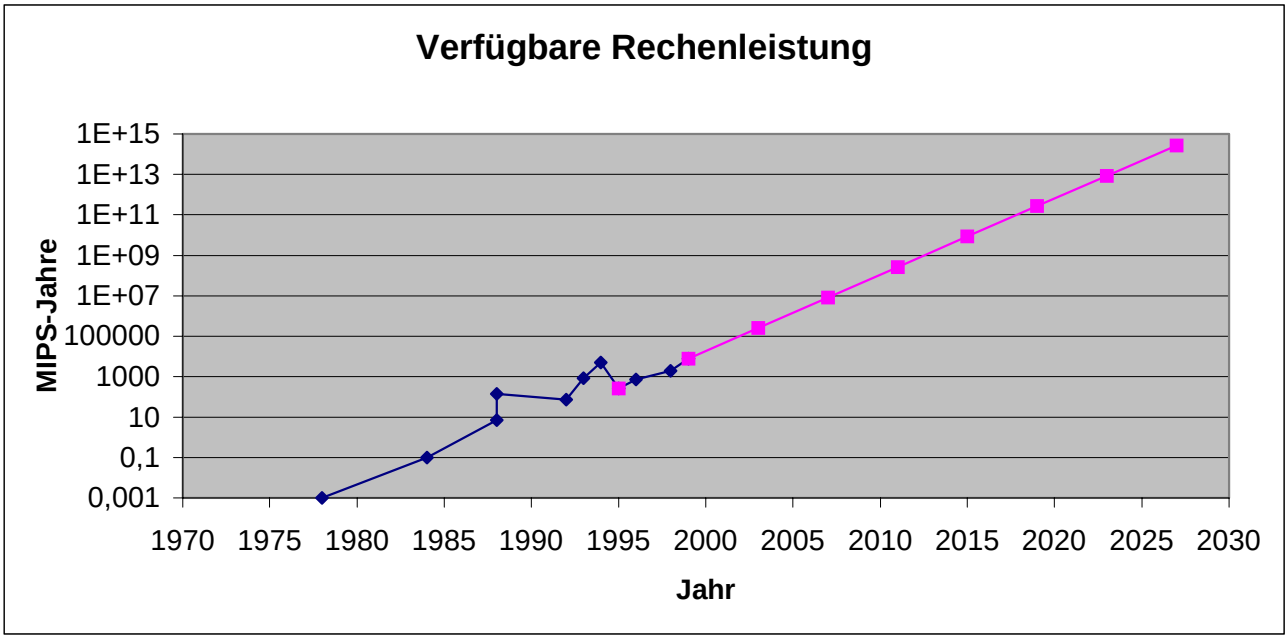

Bild 1: Abschätzung der für einen (öffentlichen) Angriff verfügbaren und nutzbaren Rechenleistung in MIPS-Jahren

Diese Schätzung liegt etwas niedriger als die sehr optimistische Schätzung von Odlyzko aus dem Jahr 1995: Unter der Annahme, dass 0,1% aller über das Internet erreichbaren Ressourcen für die Faktorisierung **einer** einzigen Zahl verwendet werden können, kommt er für das Jahr 2014 auf eine mögliche Rechenleistung von 10^{11} bis 10^{13} MIPS-Jahren. Für die für einen geheimen Angriff einer großen Organisation zur Verfügung stehende Rechenleistung kommt er hingegen ebenfalls auf 10^{10} bis 10^{11} MIPS-Jahre.

Jahr	Leistung eines PC	Geheimer Angriff	Verteilter Angriff
2004	10^3 MIPS	10^8 MJ	$2 \cdot 10^9$ MJ
2014	10^4-10^5 MIPS	10^{10}-10^{11} MJ	10^{11}-10^{13} MJ

Tabelle 1: Abschätzung der verfügbaren Ressourcen für Faktorisierungsangriffe in MIPS-Jahren (MJ) [Odly_95]

Legt man diese Abschätzungen zu Grunde, dann lässt sich die Faktorisierung großer Moduln für die kommenden zwanzig Jahre wie in Bild 2 dargestellt prognostizieren.

Danach ist frühestens im Jahr 2020 die Faktorisierung eines 1024 bit langen Moduls durch erhebliche über das Internet konzentrierte Rechenleistung zu erwarten. Erst für das Jahr 2027 muss mit der Möglichkeit zur Faktorisierung eines 1280 bit langen Moduls gerechnet werden.

Hingegen ist die Entdeckung eines im Aufwand polynomialen Faktorisierungsverfahrens nicht nur sehr unwahrscheinlich; die Existenz eines solchen Verfahrens wird von vielen Experten bezweifelt. Ein solcher "Durchbruch" der Kryptoanalyse beträfe zudem auch deutlich größere Schlüssellängen als die derzeit empfohlenen. Hiergegen ist eine Vorbeugung durch die Verwendung längerer Schlüssel ohnehin aussichtslos.

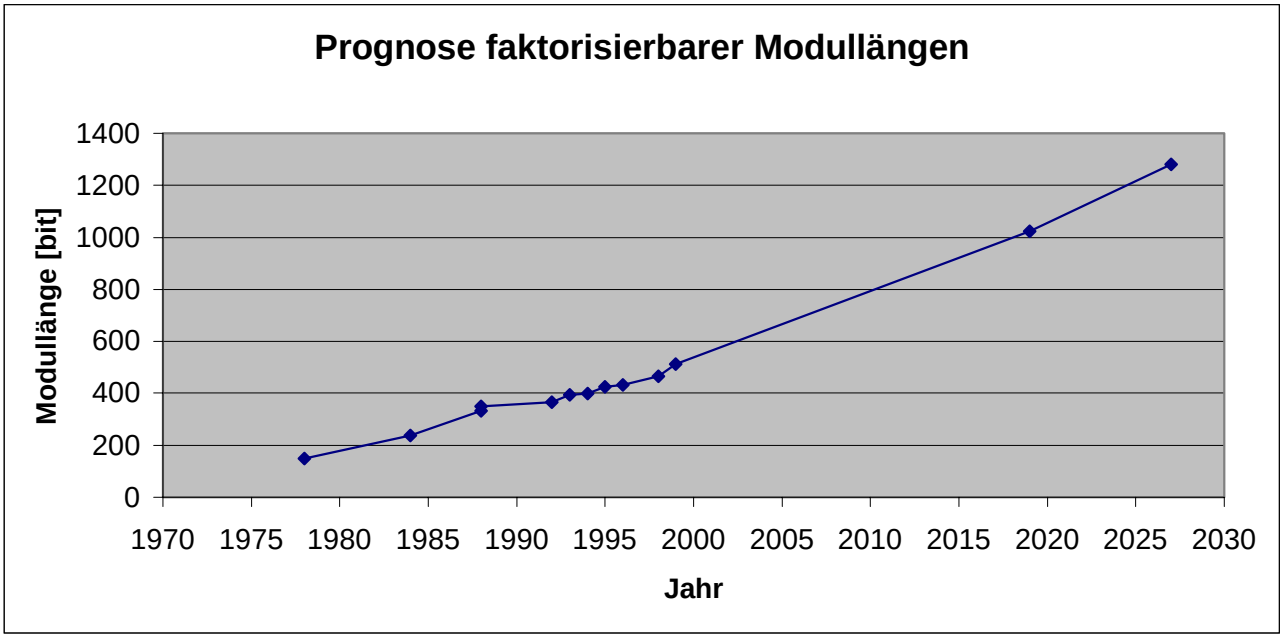

Bild 2: Prognose faktorisierbarer Modullängen [BoFT_02]

Die in Abschnitt 3.1 ausgesprochene Empfehlung, für eine Schlüsselgültigkeit bis mindestens Ende 2006 einen Schlüssel der Länge 2048 bit zu wählen, ist daher aus kryptografischer Sicht nicht begründbar. Die Empfehlung einer Schlüssellänge von 1280 bit (Mindestlänge für eine Gültigkeit bis Ende 2007) genügt hier vollständig. Für die Zeit bis Ende 2007 ist die Empfehlung einer über 1280 bit hinausgehenden Schlüssellänge (1536 bit) ebenfalls nicht erforderlich. Wir empfehlen daher, die Schlüssellängen wie folgt festzulegen:

Zeitraum / Parameter	Bis Ende 2005	Bis Ende 2006	Bis Ende 2007
N	1024	1024 (Mindestwert) 1280 (Empfehlung)	1280 (Mindestwert)

Tabelle 2: Empfehlung für minimale Bitlängen (RSA)

Diese Empfehlung deckt sich mit den Abschätzungen von Lenstra/Verheul: Dort wird eine Modullänge von 1280 bit für RSA sogar erst für das Jahr 2008 empfohlen [LeVe_99].

Der aktuelle Forschungsansatz von Bernstein [Bern_01], nach dem es für sehr große Schlüssellängen möglich sein könnte, die Aufwände mittels Spezialhardware so zu reduzieren, dass um den Faktor drei größere Schlüssellängen faktorisiert werden könnten, stellt nach weitgehend einhelliger Meinung führender Kryptologen keine Gefährdung für heute verwendete Schlüssellängen dar: Zum einen basiert das Resultat auf einer unüblichen Metrik, die – auf „klassische“ Faktorisierungsverfahren übertragen – noch bessere Ergebnisse liefert [LSTT_02]. Zum anderen reduziert sich der Aufwand nur asymptotisch, was bei kleinen Schlüssellängen wie 1024 oder 1280 bit höchstwahrscheinlich keine messbaren praktischen Auswirkungen haben dürfte, da komplexitätstheoretisch vernachlässigbare Aufwände (z.B. konstante oder polynomiale) in der Praxis meist überwiegen.

Eine Folgerung von [LSTT_02] ist, dass Empfehlungen für die RSA-Schlüssellänge, die wie [LeVe_99] allein auf der Anzahl der zur Faktorisierung benötigten Operationen beruhen, vom Ansatz von Bernstein nicht betroffen sind. Hierzu könnte man jedoch einwenden, dass sich

die in [LeVe_99] betrachteten Operationen auf einen Pentium II oder vergleichbaren Prozessor beziehen, während [Bern_01] kleinste Prozessorzellen in einer massiv-parallelen Architektur betrachtet.

Daher haben wir die Berechnungen von [LeVe_99] für dieses geänderte Architekturmodell modifiziert. Die zugrundeliegenden Parameter wurden in Anlehnung an [GeSt_02] wie folgt angepasst:

- Die Anzahl der „Impossible MIPS Years“ wurde für jedes betrachtete Jahr um den Faktor 300 erhöht, um zu erfassen, dass in der Bernstein-Architektur einzelne Operationen „billiger“ zu haben sind. Der Faktor ergibt sich aus dem Quotienten von ca. 7,5 Millionen Prozessoren eines Pentium II und ca. 2500 Transistoren, die [GeSt_02] für eine Prozessorzelle ansetzen.
- Als Vergleichsmaßstab für die benötigte Anzahl Operationen dient nicht die tatsächliche Faktorisierung der 512-Bit-Zahl RSA-155 im Jahr 1999, sondern die hypothetische Faktorisierung einer 512-Bit-Zahl mit der von [GeSt_02] beschriebenen Maschine im Jahr 2002. Hierdurch wird erfasst, dass die einzelnen Operationen in der Bernstein-Architektur tendenziell weniger „mächtig“ sind, als die bei der Faktorisierung von RSA-155 zugrunde gelegten Operationen.
- Der Parameter α der „L-Funktion“, die den asymptotischen Aufwand des Zahlkörpersiebs in der Bernstein-Architektur beschreibt, wurde auf 1,9760518 gesetzt [LSTT_02].

Wie sich in der folgenden Tabelle zeigt, ergeben sich bei dieser Betrachtungsweise sogar geringfügig kürzere Schlüssellängen als beim unmodifizierten Ansatz von [LeVe_99] empfohlen.

Jahr	2002	2003	2004	2005	2006	2007	2008	2009	2010
Empfohlene Schlüssellänge nach [LeVe_99]	1028	1068	1108	1149	1191	1235	1279	1323	1369
Angepasste Berechnung	1001	1039	1077	1117	1157	1198	1240	1283	1327

Tabelle 3: Angepasste Berechnung der minimalen Bitlänge (RSA)

Abschließend wird aus der Algorithmenempfehlung nicht ersichtlich, warum für eine potenzielle „Implementierung in der sicheren Signaturerstellungseinheit“ Abweichungen der Mindest-Bitlänge zugelassen bzw. vorgesehen werden sollten. An dieser Stelle fehlen sowohl eine Begründung für die erlaubte Abweichung als auch eine Konkretisierung.

2.4.2 DSA (Abschnitt 3.2)

Für den DSA gelten die im vorangegangenen Abschnitt ausgeführten Argumente analog, da die Algorithmen zur Berechnung diskreter Logarithmen mathematisch denjenigen zur Faktorisierung großer Zahlen verwandt sind. Nach einhelliger Kryptologenmeinung gilt der DSA als mindestens so sicher wie ein RSA-Modul, und die erfolgreichen praktischen Bestimmungen diskreter Logarithmen mit konzentrierter Rechenleistung betreffen kleinere Moduli als die bekannten RSA-Faktorisierungen. Auch Lenstra/Verheul betrachten in ihrer Analyse RSA- und DSA-Module einheitlich [LeVe_99].

Wir empfehlen daher, die Modullängen für DSA auf dieselben Werte festzulegen wie für RSA:

Zeitraum / Parameter	Bis Ende 2005	Bis Ende 2006	Bis Ende 2007
p	1024	1024 (Mindestwert) 1280 (Empfehlung)	1280 (Mindestwert)

Tabelle 4: Empfehlung für minimale Bitlängen (DSA)

2.4.3 DSA Varianten auf Gruppen (Abschnitt 3.3)

Für die Sicherheit der DSA-Varianten basierend auf Gruppen $E(F_p)$ scheint eine Sicherheit von $ord(P) = q$ von 160 bit bis Ende 2007 für ausreichend. Die Lenstra-Verheul Studie [Le-Ve_99] kommt zu dem Ergebnis, dass unter der Annahme, dass in der Kryptoanalyse der ECC-Verfahren kein Fortschritt gemacht wird, bis zum Jahr 2008 eine Schlüssellänge von 144 bit geeignet ist. Unterstellt man einen Fortschritt in der Kryptoanalyse, bei dem sich der Aufwand für einen Angreifer alle 18 Monate halbiert[2], so ist im Jahr 2008 auch eine Schlüssellänge von 155 bit noch ausreichend.

Gegen eine Verlängerung der Mindestschlüssellänge auf 180 bit spricht weiterhin, dass die DSA-Varianten über $E(F_p)$ als Hashfunktion den SHA-1 Hash mit 160 bit Hashwert verwenden. Dieser bietet eine Kollisionssicherheit von lediglich 2^{-80} und begrenzt damit die Gesamtsicherheit des Verfahrens (vgl. Kommentar zu Abschnitt 2).

Sofern zukünftig eine erhöhte Schlüssellänge von 180 bit empfohlen wird, so sollte ebenfalls eine Hashfunktion mit mindestens 180 bit Hashwert – wie z. B. der kürzlich standardisierte SHA-256 [NIST_02] – verwendet werden, da anderenfalls die Sicherheit des gesamten Signaturverfahrens nicht steigt.

2.5 Erzeugung von Zufallszahlen (Abschnitt 4)

Die Empfehlungen zur Erzeugung von Zufallszahlen sind generell zu vage gefasst und sollten – insbesondere im Hinblick auf die häufig unterschätzte Bedeutung von Zufallszahlengeneratoren – ergänzt und konkretisiert werden.

Daher empfehlen wir, zunächst die Zufallszahlen- und die Pseudozufallszahlen-Generatoren funktional zu beschreiben und dann die Anforderungen an diese zu formulieren. Anschließend sollte genauer differenziert werden, welche der Anforderungen verbindlich sind und welche den Charakter einer Empfehlung haben sowie an welche der Mechanismen (etwa Schlüsselgenerierung, DSA Signaturgenerierung) sie gestellt werden.

Es fällt auf, dass bei der Beschreibung der Pseudozufallszahlengeneratoren die Quelle (Seed) zwar erwähnt wird, jedoch keine expliziten Anforderungen an diese formuliert werden. Insbesondere fehlt bei der Beschreibung der Anforderung K3-DRNG das wichtige Kriterium der Entropie der Seed-Erzeugung zugrundeliegenden Zufallsquelle. In AIS 20 werden für verschiedene Mechanismenstärken (hoch, mittel) jeweils unterschiedliche untere Entropieschranken gefordert.

---

2 Seit Erstellung der Studie im November 1999 sind bereits 35 Monate vergangen, in denen die angenommene Halbierung der Aufwände ausgeblieben ist, in sofern kann die Annahme durchaus als konservativ gewertet werden.

Dieses Kriterium sollte explizit in die Anforderungen an Pseudozufallszahlen jeweils für die geforderte Mechanismenstärke aufgenommen werden. Es sollte zudem darauf hingewiesen werden, dass der Einsatz eines K3- oder K4-DRNG evaluierten PRNG nicht ausreichend sein könnte, denn gemäß [AIS_99] ist die Beurteilung der Seed-Generierung nicht Gegenstand der eigentlichen DRNG-Evaluation und wird von den Evaluationskriterien nicht abgedeckt.

Für die Schlüsselerzeugung wird in Abschnitt 4 gefordert, dass stets ein physikalischer Zufallszahlengenerator verwendet werden sollte. Dabei ist nicht klar, ob diese Regelung damit verbindlich ist (gemäß RFC 2119 ein „MUST“ oder „SHALL“) oder eher eine (dringende) Empfehlung darstellt (gemäß RFC 2119 ein „SHOULD“), von der ein Hersteller jedoch abweichen kann.[3]

Ist eher eine dringende Empfehlung gemeint, so muss beschrieben werden, welche Eigenschaften für die Schlüsselerzeugung verbindlich gefordert werden, z.B. die Verwendung eines K4-DRNG mit Mechanismenstärke „hoch“. Andernfalls ist zu beachten, dass die Verfügbarkeit von guten physikalischen Zufallsgeneratoren nicht generell vorausgesetzt werden kann und daher ein solcher auch nicht als Mindestanforderung verlangt werden sollte.

Bei den Anwendungen von DRNG sind insbesondere die Auswirkungen einer Kompromittierung des internen Zustands zu berücksichtigen. In diesem Sinne sind bei Anwendungen, bei denen

- der interne Zustand des DRNG am selben Ort und gegen Auslesen geschützt mit denselben Mechanismen gespeichert ist, wie der geheime Schlüssel[4] (z.B. DSA-Signaturen in einer Chipkarte) oder
- der interne Zustand des DRNG nicht persistent gespeichert wird (z.B. die einmalige Generierung eines einzelnen Schlüsselpaars),

die Anforderungen der Funktionsklasse K3 nach AIS 20 als ausreichend zu betrachten [AIS_99].

Für andere Anwendungen (z.B. die fortgesetzte Generierung von Schlüsselpaaren, bei denen der private Schlüssel an anderer Stelle als der interne Zustand des DRNG gespeichert wird und besser geschützt ist) sollte K4 als verbindlich gefordert werden. Falls der Zeitpunkt einer möglichen Kompromittierung des DRNG-Zustands ermittelt werden kann (z.B. durch manipulationserkennende Hardware), bleibt so auch bei einem erfolgreichen Einbruch die Sicherheit von in der Vergangenheit generierten Schlüsseln bzw. Signaturen gewährleistet.

Neben einigen explizit genannten Verwendungen für Zufallszahlen im Umfeld von Signaturen ist in der Empfehlung vage von „anderen Anwendungen“ die Rede. In Verfolgung eines konservativen Ansatzes sollten für diese unspezifizierten Anwendungen die höchsten auch an anderer Stelle verlangten Anforderungen gestellt werden, sofern diese nicht näher spezifiziert werden können.

Auch an einigen anderen Stellen sind die Formulierungen zu unspezifisch gewählt worden und sollten den konkreteren Beschreibungen der ersten Abschnitte angepasst werden. Beispielsweise wird gefordert, auf einen Rauschalarm „angemessen“ zu reagieren, ohne dies näher zu definieren. Auch eine Formulierung wie „Bei Bedarf kann [...] auf das Know-how des BSI zurückgegriffen werden.“ ist einem Dokument dieser Art eher unangebracht.

---

3 Generell wird empfohlen, die Begrifflichkeiten (soll, muss) in der Einleitung des Dokuments einzuführen und zu erläutern.

4 Bzw. andere Werte, deren Sicherheit von der Sicherheit des DRNG abhängt.

Für eine bessere Übersichtlichkeit könnten die Anforderungen analog zum Abschnitt 3 tabellarisch zusammengefasst werden. Unter Berücksichtigung der obigen Kommentare ergäbe sich folgende Darstellung:

Anwendung	**Mindestanforderung**	**Empfehlung**
Generierung eines Schlüsselpaars (Interner Zustand wird persistent gespeichert & Schutz des internen Zustandes nicht wie für den geheimen Schlüssel)	K4-DRNG/hoch	Physikalischer RNG
Signaturparameter *k* bei DSA-Varianten (Interner Zustand wird persistent gespeichert & Schutz des internen Zustandes nicht wie für den geheimen Schlüssel)		
Generierung eines Schlüsselpaars (Interner Zustand wird nicht persistent gespeichert oder Schutz des internen Zustandes wie für den geheimen Schlüssel)	K3-DRNG/hoch	Physikalischer RNG
Signaturparameter k bei DSA-Varianten (Interner Zustand wird nicht persistent gespeichert oder Schutz des internen Zustandes wie für den geheimen Schlüssel)		
Andere (unspezifiziert)	K4-DRNG/ (extra-) hoch	Physikalischer RNG

Tabelle 5: Anforderungen an Zufallszahlengeneratoren

3 Literatur

A9C_02 *Minutes of the meeting of the electronic signature committee*, Brüssel, 08.07.2002.

AIS_99 AIS 20: *Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren*. Version 2, 02.11.1999

BaRi_00 Barreto, P.S.L.M., Rijmen, V.: *The WHIRLPOOL Hashing Function*, 2000, https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/whirlpool.zip

Bern_01 Bernstein, Daniel J.: *Circuits for Integer Factoriszation: A Proposal*. 09.11.2001, http://cr.yp.to/papers.html#nfscircuit.

BeBF_02 Bertsch, Andreas; Bourseau, Frank; Fox, Dirk: *Perspektive kryptografischer Verfahren auf elliptischen Kurven*. Datenschutz und Datensicherheit (DuD), 2/2002 , S. 90-96.

BoFT_02 Bourseau, Frank; Fox, Dirk; Thiel, Christoph: *Vorzüge und Grenzen des RSA-Verfahrens*. Datenschutz und Datensicherheit (DuD), 2/2002, S. 84-89.

BuLP_93 Buhler, J.P.; Lenstra, H.W.; Pomerance, C.: *Factoring integers with the number field sieve*. In: Lenstra, A.K.; Lenstra, H.W. (Hrsg.): The Development of the

Number Field Sieve. Lecture Notes in Mathematics, Vol. 1554, Springer, Heidelberg 1993, S. 50-94.

FKMP_02 Fox, Dirk; Knobloch, Hans-Joachim; Michels, Markus; Petersen, Holger: *Stellungnahme zur Empfehlung „Geeignete Kryptoalgorithmen“ gemäß §17 (1) SigG (Update 2002)*. 07.03.2002.

GeSt_02 Geiselmann, Willi; Steinwandt, Rainer: *A Dedicated Sieving Hardware,* wird veröffentlich in Public Key Cryptography PKC 2003.

LeVe_99 Lenstra, Arjen K.; Verheul, Eric: *Selecting Cryptographic Key Sizes*. November 24, 1999; http://www.cryptosavvy.com.

LSTT_02 Lenstra, A.K.; Shamir, A.; Tomlinson, J.; Tromer, E.: *Analysis of Bernstein's Factorization Circuit*, 2002; http://www.cryptosavvy.com.

NIST_00 National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186-2 (FIPS-PUB), 27.01.2000.

NIST_02 NIST: *FIPS 180-2: Secure Hash Standard (SHS)*, 01.08.2002, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

Odly_95 Odlyzko, Andrew M.: *The Future of Integer Factorisation*. Cryptobytes, Summer 1995, Vol. 1, No. 2, S. 5-12.