# Empfehlung „Geeignete Kryptoalgorithmen" gemäß §17 (1) SigG (Update 2002)
## Stellungnahme

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

Version 1.0
Stand 07. März 2002

## Inhaltsübersicht

## Abkürzungen

| | |
|---|---|
| ANSI | American National Standards Institute |
| DRNG | Deterministic RNG |
| DSA | Digital Signature Standard |
| 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 |
| 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 des „Update 2002“ der Empfehlung 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.

# 2 Kommentierung des Empfehlungsentwurfs vom 24.01.2002

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

## 2.1 Kryptographische Anforderungen (Abschnitt 1)

Keine Kommentare.

## 2.2 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 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) die Mindestlänge des Schlüsselparameters *q* von 160 bit auf 180 bit erhöht werden. Dadurch divergieren die vorher aufeinander abgestimmten Bitlängen von *q* und dem Ausgabewert der Hashfunktion, was Fragen hinsichtlich der erreichten Sicherheit aufwirft. Momentan gibt es keine etablierten Alternativen zu den genannten Hashfunktionen RIPEMD-160 und SHA-1.

So bleibt als Empfehlung lediglich, die aktuellen Entwicklungen von Hashfunktionen mit mehr als 160-bit Ausgabe zu verfolgen und ggf. voranzutreiben. Die aussichtsreichsten Kandidaten sind derzeit:

- Die Vorschläge von NIST, bei einer Revision von FIPS-180 zusätzlich zum existierenden SHA-1 auch 256-bit bis 512-bit Hashfunktionen SHA-256, SHA-384 und SHA-512 zu standardisieren.
- 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.

## 2.3 Vorschläge für geeignete Signaturalgorithmen (Abschnitt 3)

### 2.3.1 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 sprechen die folgenden Feststellungen und Überlegungen:

- Mit dem RSA-Verfahren wurde 1978 erstmals eine bahnbrechende praktische Anwendung der Zahlentheorie entdeckt. Dadurch gewann die Untersuchung des Faktorisierungsproblems erhebliche wissenschaftliche Aufmerksamkeit und in direkter Folge starken Zulauf. Erhebliche Verbesserungen der Faktorisierungsverfahren gelangen vor allem in den anschließenden zwölf Jahren.
- 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. Alle seitdem veröffentlichten Vorschläge betrafen Verbesserungen des Algorithmus. Mit dem "General Number Field Sieve" (GNFS) fand die theoretische Weiterentwicklung der Faktorisierungsalgorithmen 1993 einen bis heute gültigen Abschluss [BuLP_93].
- Die Implementierung und Nutzung des GNFS-Algorithmus zur Faktorisierung großer Zahlen erfolgte erstmals 1995. Seither verlief die Entwicklung der Faktorisierungserfolge bis zum Jahr 1999 (Faktorisierung eines 512-bit-Moduls) auffallend linear (siehe Abbildung) [BoFT_02]. Diese Faktorisierungserfolge waren fast ausschließlich auf die Steigerung der Leistungsfähigkeit verfügbarer Rechensysteme und die Bildung von "virtuellen Parallelrechnern" (Rechen-Clustern) aus über das Internet verbundenen Einzelsystemen zurückzuführen.

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. In diese Richtung zielt auch der Forschungsansatz von Bernstein [Bern_01], der den Aufwand zur Faktorisierung unter Verwendung einer speziellen Hardware für große Modullängen asymptotisch deutlich reduziert.

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.

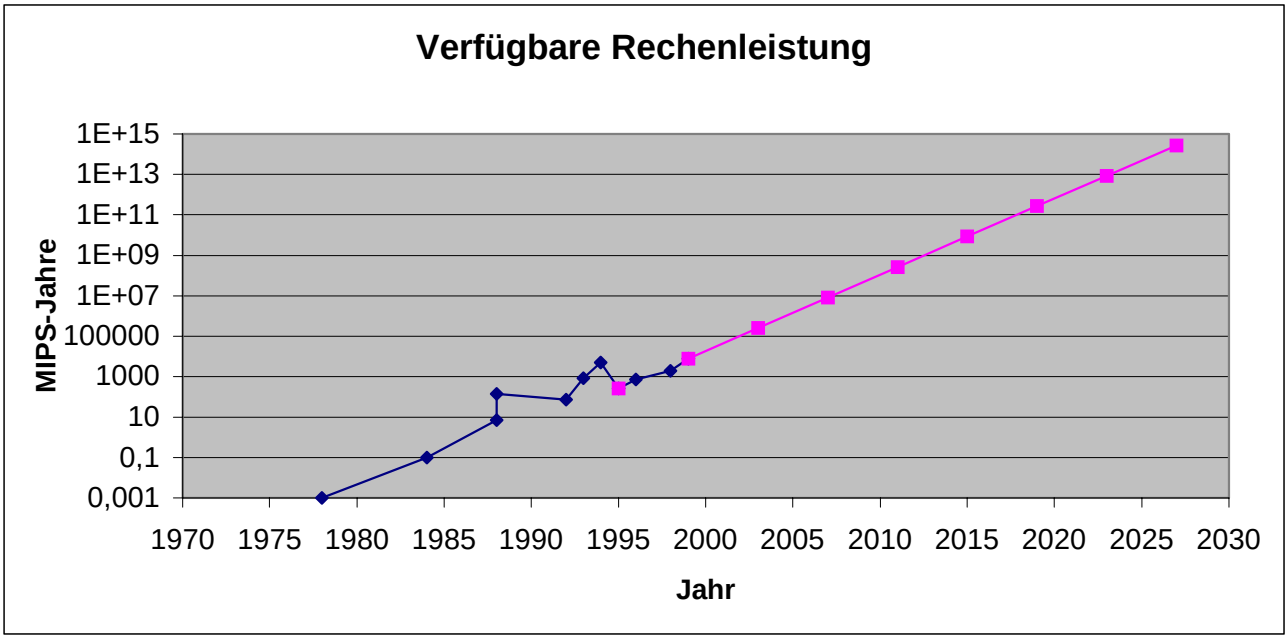


Bild 1: Abschätzung der für einen (öffentlichen) Angriff verfügbaren und nutzbaren Rechenleistung in MIPS-Jahren

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 mit einer Faktorisierung eines 1024 bit langen Moduls zu rechnen – mit einem organisatorischen Aufwand und Bedarf an konzentrierter Rechenleistung, der in Relation dem für die Faktorisierung eines 512 bit Moduls im Jahr 1999 benötigten vergleichbar ist. Erst für das Jahr 2027 muss danach mit der Möglichkeit zur Faktorisierung eines 1280 bit langen Moduls gerechnet werden.

Hingegen ist die Entdeckung eines im Aufwand polynomiellen 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. Daher ist hiergegen eine Vorbeugung durch die Verwendung längerer Schlüssel wenig aussichtsreich.

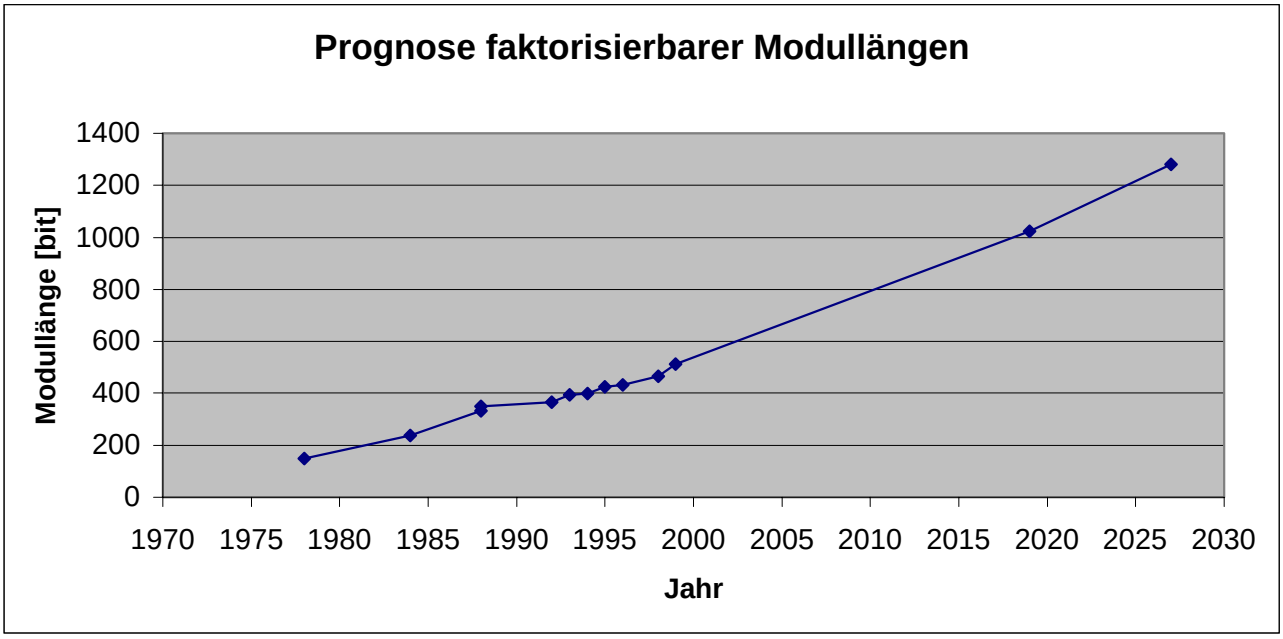


Bild 2: Prognose faktorisierbarer Modullängen [BoFT_02]

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 ebenfalls nicht erforderlich.

Diese Empfehlung erscheint auch unter Berücksichtigung der Abschätzungen von Lenstra/Verheul sinnvoll: Dort wird eine Modullänge von 1280 bit für RSA sogar erst für das Jahr 2008 empfohlen [LeVe_99].

| Zeitraum / Parameter | Bis Ende 2005 | Bis Ende 2006 | Bis Ende 2007 |
|---|---|---|---|
| N | 1024 | 1024 (Mindestwert)<br>1280 (Empfehlung) | 1280 (Mindestwert) |

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 unserer Einschätzung derzeit für die oben gegebene Empfehlung keine Gefährdung dar: Zum einen gibt es die erforderliche spezielle Hardware derzeit nicht. Zum anderen reduziert sich der Aufwand nur asymptotisch, was bei kleinen Bitlängen wie 1024 oder 1280 nicht notwendig messbare praktische Auswirkungen haben muss, da komplexitätstheoretisch vernachlässigbare Aufwände (z.B. konstante oder polynomiale) in der Praxis überwiegen könnten. In jedem Fall sind die Arbeiten von Bernstein ein ernstzunehmender Forschungsansatz, dessen Ergebnisse hinsichtlich ihrer praktischen Bedeutung für zukünftige Schlüssellängen aufmerksam verfolgt werden sollten.

### 2.3.2 DSA (Abschnitt 3.2)

Neben direkten Sicherheitserwägungen und Abschätzungen ist bei einer Empfehlung von Kryptoalgorithmen auch die Standardkonformität zu beachten. Diese wird durch den Mindestwert für *p* ab 2007 (180 bit) verletzt, da im DSS eine Länge von 160 bit festgeschrieben ist [NIST_00]. Dies dürfte insbesondere im internationalen Umfeld zu Interoperabilitätsproblemen führen.

Es wird daher empfohlen, das NIST zu überzeugen, den DSS-Standard FIPS 186 insoweit zu modifizieren, dass in der nächsten Version des Standards auch höhere Werte für *p* zugelassen werden.

### 2.3.3 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 [LeVe_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 alle 18 Monate halbiert[1], 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 $2^{-80}$ 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 verwendet werden, da ansonsten die Sicherheit des gesamten Signaturverfahrens nicht steigt.

## 2.4 Erzeugung von Zufallszahlen (Abschnitt 4)

Die Empfehlungen zur Erzeugung von Zufallszahlen sind zu vage gefasst.

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 der Seed-Erzeugung zugrundeliegenden Zufallsquelle. In AIS 20 werden für verschiedene Mechanismenstärken (hoch, mittel) jeweils unterschiedliche untere Entropieschranken gefordert.

Dieses Kriterium sollte explizit in die Anforderungen an Pseudozufallszahlen jeweils für die geforderte Mechanismenstärke aufgenommen werden. Es sollte zudem darauf hingewiesen

---

1 Seit Erstellung der Studie im November 1999 sind bereits 28 Monate vergangen, in denen die angenommene Halbierung der Aufwände ausgeblieben ist, in sofern kann die Annahme durchaus als konservativ gewertet werden.

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 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.[2]

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[3] (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.

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

---

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

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

| 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 |

# 3 Literatur


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.

LeVe_99 Lenstra, Arjen K.; Verheul, Eric: *Selecting Cryptographic Key Sizes*. November 24, 1999; 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_01 NIST: *Draft FIPS 180-2: Secure Hash Standard (SHS)*, 30.05.2001, http://csrc.nist.gov/encryption/shs/dfips-180-2.pdf

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