

Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen

Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen)

Vom 17. Dezember 2007

Die Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen als zuständige Behörde gemäß § 3 Signaturgesetz (SigG) vom 16. Mai 2001 (BGBl. I S. 876), zuletzt geändert durch Artikel 4 des Gesetzes vom 26. Februar 2007 (BGBl. I S. 179), veröffentlicht gemäß Anlage 1 Abschnitt 1 Nr. 2 Signaturverordnung (SigV) vom 16. November 2001 (BGBl. I S. 3074), zuletzt geändert durch Erstes Gesetz zur Änderung des Signaturgesetzes (1. SigÄndG) vom 4. Januar 2005 (BGBl. I S. 2), im Bundesanzeiger eine Übersicht über die Algorithmen und zugehörigen Parameter, die zur Erzeugung von Signaturschlüsseln, zum Hashen zu signierender Daten oder zur Erzeugung und Prüfung qualifizierter elektronischer Signaturen als geeignet anzusehen sind, sowie den Zeitpunkt, bis zu dem die Eignung jeweils gilt.

Geeignete Algorithmen zur Erfüllung der Anforderungen nach § 17 Abs. 1 bis 3 SigG in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV

Vorbemerkung: Das BSI wurde in den vergangenen Jahren von verschiedener Seite um eine etwas längerfristige Einschätzung als nur für sechs Jahre gebeten. Normalerweise sind solche längerfristigen Prognosen schwer möglich. Nachdem aber kürzlich das generelle Sicherheitsniveau von 80 auf 100 Bit angehoben wurde, ist das Sicherheitspolster gegenüber dem aktuellen Stand der Kryptoanalyse augenblicklich größer, als es in den meisten Vorjahren war. Das heißt konkret, dass alle bis Ende 2013 als geeignet betrachteten Algorithmen und Schlüssellängen aus Sicht des BSI auch bis Ende 2014 geeignet erscheinen.

Die Sicherheit einer qualifizierten elektronischen Signatur hängt primär von der Stärke der zugrunde liegenden Algorithmen ab. Im Folgenden werden Algorithmen genannt, die für qualifizierte elektronische Signaturen mindestens für die **kommenden sieben Jahre**¹ (d.h. bis Ende 2014¹) als geeignet anzusehen sind. Die bitgenauen Spezifikationen findet man in den entsprechenden Standards verschiedener Organisationen (ISO/IEC, NIST, IEEE usw.). Ebenso wie patentrechtliche Fragen und Definitionen der mathematischen Begriffe sind diese Spezifikationen nicht Gegenstand der vorliegenden Veröffentlichung. Informationen hierzu findet man in der einschlägigen Literatur (Lehrbücher, Proceedings von Konferenzen etc.) und im Internet.

¹ Siehe Vorbemerkung

In dieser Veröffentlichung werden die wichtigsten, praxisrelevanten Algorithmen betrachtet, deren kryptographische Eigenschaften aufgrund der heute vorliegenden Ergebnisse langjähriger Diskussionen und Analysen am besten eingeschätzt werden können. Die Liste dieser Algorithmen wird gemäß der weiteren Entwicklung der kryptologischen Forschung und den Erfahrungen mit praktischen Realisierungen von Signaturverfahren aktualisiert und bei Bedarf ergänzt werden.

Auf die Sicherheit einer konkreten Implementierung in Hard- und Software wird hier nicht eingegangen. Diese wird im Rahmen der Untersuchung nach § 15 Abs. 7 und § 17 Abs. 4 SigG festgestellt.

1. Kryptographische Anforderungen

Nach Anlage 1 Abschnitt I Nr. 2 SigV sind folgende Algorithmen festzulegen:

- Ein Algorithmus zum Hashen von Daten (eine Hashfunktion), der die zu signierenden Daten auf einen Hashwert, d.h. eine Bitfolge fester kurzer Länge, reduziert. Signiert werden dann nicht die Daten selbst, sondern stattdessen jeweils ihr Hashwert.
- Ein asymmetrisches Signaturverfahren, das aus einem Signieralgorithmus und einem Verifizieralgorithmus besteht. Das Signaturverfahren hängt ab von einem Schlüsselpaar, bestehend aus einem privaten (d.h. geheimen) Schlüssel zum Signieren (gemäß § 2 Nr. 4 SigG als Signaturschlüssel zum Erzeugen einer Signatur bezeichnet) und dem dazugehörigen öffentlichen Schlüssel zum Verifizieren der Signatur (gemäß § 2 Nr. 5 SigG als Signaturprüf Schlüssel zur Überprüfung einer Signatur bezeichnet).
- Ein Verfahren zur Erzeugung von Schlüsselpaaren für Signaturverfahren.

1.1. Hashfunktionen

Beim Signieren und Verifizieren wird der Hashwert der zu signierenden Daten gewissermaßen wie ein 'digitaler Fingerabdruck' benutzt. Damit hierbei keine Sicherheitslücke entsteht, muss die Hashfunktion H folgenden Kriterien genügen:

- H muss *kollisionsresistent* sein; d.h., es ist praktisch unmöglich, Kollisionen zu finden. (Zwei unterschiedliche digitale Dokumente, die auf denselben Hashwert abgebildet werden, bilden eine Kollision).
- H muss eine *Einwegfunktion* sein; d.h., es ist praktisch unmöglich, zu einem gegebenen Bitstring aus dem Wertebereich ein Urbild bzgl. H zu finden.

Die Existenz von Kollisionen ist unvermeidbar. Dies ist aber nur eine theoretische Aussage. Bei der praktischen Anwendung kommt es nur darauf an, dass es, wie oben verlangt, unmöglich ist, Kollisionen (bzw. Urbilder) zu *finden*.

1.2. Signaturverfahren

Niemand anders als der Besitzer des Signaturschlüssels darf in der Lage sein, Signaturen zu erzeugen. Insbesondere bedeutet dies, dass es praktisch unmöglich ist, den Signaturschlüssel aus dem (öffentlichen) Signaturprüfchlüssel zu berechnen.

1.3. Schlüsselerzeugung

Die verschiedenen Signaturverfahren benötigen Schlüssel mit gewissen Eigenschaften, die sich aus dem jeweiligen konkreten Verfahren ergeben. Im Folgenden werden weitere einschränkende Bedingungen festgelegt, deren Nichtbeachtung zu Schwächen führen könnte. Zusätzlich wird generell verlangt, dass Schlüssel nach den unter „4. Erzeugung von Zufallszahlen“ genannten Maßnahmen zufällig erzeugt werden.

2. Geeignete Hashfunktionen

Nach heutigem Kenntnisstand der Analyse von Hashfunktionen kann bis auf weiteres die Hashfunktion RIPEMD-160 [3] **bis Ende 2010** als geeignet für qualifizierte elektronische Signaturen betrachtet werden.

Die Hashfunktion SHA-1 [2], [3] kann grundsätzlich bis auf weiteres **bis Ende 2007** als geeignet angesehen werden; abweichend hiervon kann die Hashfunktion SHA-1 im Rahmen einer **Übergangsfrist bis Ende Juni 2008** genutzt werden. Für die Erzeugung qualifizierter Zertifikate (d.h. nicht zur Erzeugung und Prüfung anderer qualifiziert signierter Daten) kann SHA-1 bis auf weiteres **bis Ende 2009** als geeignet angesehen werden. Unter der zusätzlichen Voraussetzung, dass mindestens 20 Bit Entropie in die Generierung der Seriennummer eines qualifizierten X.509-Zertifikats eingehen, ist SHA-1 bis auf weiteres **bis Ende 2010** geeignet für die Erzeugung solcher Zertifikate.

Für die Prüfung qualifizierter Zertifikate sind SHA-1 und RIPEMD-160 **bis Ende 2014¹** geeignet.

Diese Angaben über die genannten beiden 160-Bit Hashfunktionen sind als vorläufig zu verstehen: Neue Methoden zur Kollisionssuche (insbesondere [23] sowie angekündigte Verbesserungen) können vorerst noch nicht abschließend beurteilt werden, so dass Korrekturen der Angaben (nach oben oder unten) in den nächsten Jahren gut möglich erscheinen.

Folgende Hashfunktionen mit verschiedenen Hashwert-Längen (SHA-224 ist eine 224-Bit Hashfunktion etc.) sind geeignet, ein langfristiges Sicherheitsniveau zu gewährleisten:

- SHA-224, SHA-256, SHA-384, SHA-512 [2].

Diese vier Hashfunktionen sind (mindestens) in den **kommenden sieben Jahren¹**, d.h. **bis Ende 2014¹**, für die Anwendung bei qualifizierten elektronischen Signaturen geeignet.

Die folgende Tabelle fasst die Eignung der Hashfunktionen zusammen.

¹ Siehe Vorbemerkung

geeignet bis Ende 2007 (Übergangsfrist bis Ende Juni 2008)	Erzeugung qualifizierter Zertifikate*: geeignet bis Ende 2009	Erzeugung qualifizierter Zertifikate**: geeignet bis Ende 2010	geeignet bis Ende 2010	geeignet bis Ende 2014 ¹
SHA-1	SHA-1	SHA-1	RIPEMD-160	SHA-224, SHA-256, SHA-384, SHA-512 (SHA-1, RIPEMD-160)***

*d.h. zur Erzeugung qualifizierter Zertifikate, nicht aber zur Erzeugung und Prüfung anderer qualifiziert signierter Daten.

** d.h. zur Erzeugung qualifizierter Zertifikate bei ≥ 20 Bit Entropie der Seriennummer, nicht aber zur Erzeugung und Prüfung anderer qualifiziert signierter Daten.

***ausschließlich zur Prüfung qualifizierter Zertifikate.

3. Geeignete Signaturverfahren

Im Jahr 1977 haben Rivest, Shamir und Adleman als Erste ein Verfahren zum Erzeugen und Verifizieren digitaler Signaturen explizit beschrieben. Es handelt sich um das nach seinen Erfindern benannte RSA-Verfahren [9]. Im Jahr 1984 hat ElGamal [8] ein weiteres Signaturverfahren vorgeschlagen. Eine Variante dieses ElGamal-Verfahrens ist der 1991 vom National Institute of Standards and Technology (NIST) publizierte Digital Signature Standard (DSS) [1], der den Digital Signature Algorithm (DSA) spezifiziert. Daneben gibt es Varianten des DSA, die auf Punktgruppen $E(K)$ elliptischer Kurven über endlichen Körpern K basieren, wobei

$K = F_p$ ein endlicher Primkörper bzw. $K = F_{2^m}$ ein endlicher Körper der Charakteristik 2 ist.

Folgende Signaturverfahren sind zur Erfüllung der Anforderungen nach § 17 Abs. 1 bis 3 SigG geeignet:

1. RSA-Verfahren [4],
2. DSA [1], [4],
3. DSA-Varianten, basierend auf elliptischen Kurven. Insbesondere die Verfahren:
 - EC-DSA [1], [5], [10], [11],
 - EC-KDSA, EC-GDSA [11],
 - Nyberg-Rueppel-Signaturen [19], [6].

Die Sicherheit der oben genannten Verfahren beruht dabei entsprechend auf:

1. dem Faktorisierungsproblem für ganze Zahlen,
2. dem Diskreten-Logarithmus-Problem in der multiplikativen Gruppe eines Primkörpers F_p ,
3. dem Diskreten-Logarithmus-Problem in den Gruppen $E(F_p)$ bzw. $E(F_{2^m})$.

¹ Siehe Vorbemerkung

Um festzulegen, wie groß die Systemparameter bei diesen Verfahren gewählt werden müssen, um deren Sicherheit zu gewährleisten, müssen zum einen die besten heute bekannten Algorithmen zum Faktorisieren ganzer Zahlen bzw. zum Berechnen diskreter Logarithmen (in den oben genannten Gruppen) betrachtet und zum anderen die Leistungsfähigkeit der heutigen Rechnertechnik berücksichtigt werden. Um eine Aussage über die Sicherheit für einen bestimmten zukünftigen Zeitraum zu machen, muss außerdem eine Prognose für die beiden genannten Aspekte zugrunde gelegt werden, vgl. [13]. Solche Prognosen sind nur für relativ kurze Zeiträume sinnvoll (und können sich natürlich jederzeit aufgrund unvorhersehbarer dramatischer Entwicklungen als falsch erweisen).

Im Folgenden bezeichnen wir mit der Bitlänge r einer Zahl $x > 0$ diejenige ganze Zahl r mit der Eigenschaft $2^{r-1} \leq x < 2^r$.

Die Sicherheit der einzelnen Verfahren ist (mindestens) für die **kommenden sieben Jahre**¹, d.h. bis **Ende 2014**¹, bei der im Folgenden festgelegten Wahl der Parameter gewährleistet.

3.1. RSA-Verfahren

Für den zugrunde liegenden Modulus $n=pq$ (p und q Primzahlen) reicht **bis Ende 2007** eine Länge von 1024 Bit aus; abweichend hiervon kann die Länge von 1024 Bit im Rahmen einer **Übergangsfrist bis Ende März 2008** genutzt werden. Für den Zeitraum **bis Ende 2008** wird eine Länge von 1280 Bit als ausreichend sicher angesehen, **bis Ende 2009** muss n eine Bitlänge von mindestens 1536 haben, **bis Ende 2010** mindestens 1728 Bit und **ab Anfang 2011** mindestens 1976 Bit. Für die Gewährleistung eines langfristigen Sicherheitsniveaus wird die Erhöhung auf 2048 Bit empfohlen.

Die folgende Tabelle fasst die minimalen Bitlängen zusammen.

Zeitraum Parameter	bis Ende 2007 (Übergangsfrist bis Ende März 2008)	bis Ende 2008	bis Ende 2009	bis Ende 2010	bis Ende 2014 ¹
n	1024 (Mindestw.) 2048 (Empf.)	1280 (Mindestw.) 2048 (Empf.)	1536 (Mindestw.) 2048 (Empf.)	1728 (Mindestw.) 2048 (Empf.)	1976 (Mindestw.) 2048 (Empf.)

Die Primfaktoren p und q von n sollten die gleiche Größenordnung haben, aber nicht zu dicht beieinander liegen:

$$\varepsilon_1 < |\log_2(p) - \log_2(q)| < \varepsilon_2.$$

Als Anhaltspunkte für die Werte ε_1 und ε_2 werden hier $\varepsilon_1 \approx 0,1$ und $\varepsilon_2 \approx 30$ vorgeschlagen. Die Primfaktoren p und q müssen unter Beachtung der genannten Nebenbedingungen zufällig und unabhängig voneinander erzeugt werden.

¹ Siehe Vorbemerkung

Der öffentliche Exponent e wird unabhängig von n unter der Nebenbedingung $\text{ggT}(e, (p-1)(q-1)) = 1$ gewählt. Der zugehörige geheime Exponent d wird dann in Abhängigkeit von dem vorher festgelegten e berechnet, so dass $ed \equiv 1 \pmod{\text{kgV}(p-1, q-1)}$ gilt.

Bemerkungen:

- Die Forderung, dass p und q *starke* Primzahlen sein müssen (d. h. $p-1$ und $q-1$ haben große Primfaktoren etc.), erscheint im Hinblick auf die heute bekannten besten Faktorisierungsalgorithmen nicht mehr ausreichend begründet und daher verzichtbar.
- Der öffentliche Exponent e kann zufällig gewählt werden. Auf der anderen Seite haben kleine öffentliche Exponenten den Vorteil, dass die Verifikation einer Signatur sehr schnell durchgeführt werden kann. Das hier verlangte Verfahren (zuerst Wahl von e , danach Wahl von d) soll gewährleisten, dass kleine geheime Exponenten ausgeschlossen werden können, siehe z.B. [20].
- Der Hashwert muss vor der Anwendung des geheimen Exponenten d auf die Bitlänge des Moduls formatiert werden. Das Formatierungsverfahren ist dabei sorgfältig zu wählen, siehe [14]. Geeignete Verfahren sind zum Beispiel:
 - RSA: „Signature Schemes with Appendix“ PSS aus [15] Abschn. 8.1 und 9.1,
 - „DSI according to ISO/IEC 9796-2 with random number“ [16],
 - „digital signature scheme 2“ und „digital signature scheme 3“ aus [21].
 - Darüber hinaus ist das Formatierungsverfahren RSA: „Signature Schemes with Appendix“ PKCS1-v1_5 aus [15] Abschn. 8.2 und 9.2 noch bis Ende 2014 geeignet. Es wird aber empfohlen, dieses Verfahren nicht über Ende 2013 hinaus zu verwenden.
- Die Realisierung eines Formatierungsverfahrens – z. B. die Form der Arbeitsteilung zwischen einer Chipkarte, auf der die Potenzierung mit dem geheimen Schlüssel durchgeführt wird, und dem Hintergrundsystem – ist für die Sicherheit durchaus relevant und muss im Rahmen der Prüfung technischer Komponenten nach § 15 Abs. 7 und § 17 Abs. 4 SigG untersucht werden.
- Zur Erzeugung der Primfaktoren siehe [5] und [17]. Insbesondere muss bei Nutzung eines probabilistischen Primzahltests mit hinreichender Wahrscheinlichkeit ausgeschlossen sein, dass p oder q in Wirklichkeit zusammengesetzte Zahlen sind. Als Anhaltspunkt für eine obere Schranke für diese Wahrscheinlichkeit wird hier bis Ende 2009 der Wert 2^{-80} , ab Anfang 2010 der Wert 2^{-100} (siehe [1]; vergleiche aber auch [5] und [17]) vorgeschlagen.

3.2. DSA

In FIPS-186 [1] wird für den Parameter p (p Primzahl) eine Bitlänge von genau 1024 verlangt. Gleichzeitig wird dort die Bitlänge des Parameters q auf 160 festgelegt.

Für den Zeitraum **bis Ende 2007** reicht eine Minimallänge des Parameters p von 1024 Bit aus. Abweichend von FIPS-186 wird hier verlangt, dass für den Zeitraum **bis Ende 2008** die Minimallänge des Parameters p 1280 Bit betragen muss. Für den Zeitraum **bis Ende 2009** wird die Bitlänge des Parameters p von mindestens 1536 als ausreichend sicher angesehen; **ab Anfang 2010** muss die Bitlänge von p mindestens 2048 Bit betragen. Für die Gewährleistung eines langfristigen Sicherheitsniveaus wird grundsätzlich die Erhöhung auf 2048 Bit empfohlen.

Die Bitlänge des Parameters q soll **bis Ende 2009** mindestens 160 betragen. **Ab Anfang 2010** muss die Bitlänge von q mindestens 224 betragen.

Die folgende Tabelle fasst die Bitlängen für den DSA zusammen.

Zeitraum Parameter	bis Ende 2007	bis Ende 2008	bis Ende 2009	bis Ende 2014 ¹
p	1024 (Mindestwert) 2048 (Empfehlung)	1280 (Mindestwert) 2048 (Empfehlung)	1536 (Mindestwert) 2048 (Empfehlung)	2048
q	160	160	160	224

Bemerkungen:

- Zur Erzeugung von p und der weiteren Parameter siehe [1]; ab Anfang 2010 soll aber die Wahrscheinlichkeit, dass p oder q zusammengesetzt sind, kleiner als 2^{-100} sein.
- In FIPS-186 wird die Bitlänge des Parameters q auf genau 160 festgelegt. Dies erlaubt die Konstruktion von 'Kollisionen' im Sinne von [12] bei der Parametergenerierung. Diese Kollisionen haben jedoch in der Praxis keine Bedeutung. Soll dessen ungeachtet die Möglichkeit, diese Kollisionen konstruieren zu können, ausgeschlossen werden, sind Bitlängen > 160 zu wählen.

3.2.a) DSA-Varianten basierend auf Gruppen $E(F_p)$

Um die Systemparameter festzulegen, werden eine elliptische Kurve E und ein Punkt P auf $E(F_p)$ erzeugt, so dass folgende Bedingungen gelten:

- $ord(P) = q$ mit einer von p verschiedenen Primzahl q .
- $r_0 := \min(r : q \text{ teilt } p^r - 1)$ ist groß, konkret $r_0 > 10^4$.
- Die Klassenzahl der Hauptordnung, die zum Endomorphismenring von E gehört, ist mindestens 200.

Für den Zeitraum **bis Ende 2009** muss die Bitlänge von p mindestens 192 betragen. Dabei sollte q sich nur um einen kleinen Faktor von p unterscheiden. Auf jeden Fall ist sicherzustellen, dass die Bitlänge von q mindestens 180 Bit beträgt. **Ab Anfang 2010** muss die Länge von q mindestens 224 Bit betragen, weitere Bedingungen an p werden nicht gestellt.

Die folgende Tabelle fasst die Bitlängen für DSA-Varianten basierend auf Gruppen $E(F_p)$ zusammen.

Parameter \ Zeitraum	bis Ende 2009	bis Ende 2014 ¹
p	192	
q	180	224

¹ Siehe Vorbemerkung

Bemerkung: Die untere Abschätzung für r_0 hat den Sinn, Attacken auszuschließen, die auf einer Einbettung der von P erzeugten Untergruppe in die multiplikative Gruppe eines Körpers F_{p^r} beruhen. In der Regel (bei zufälliger Wahl der elliptischen Kurve) ist diese Abschätzung erfüllt, denn r_0 ist die Ordnung von $p \pmod{q}$ in F_q^* und hat deshalb im Allgemeinen sogar dieselbe Größenordnung wie q . Im Idealfall sollte r_0 explizit bestimmt werden, was allerdings die etwas aufwändige Faktorisierung von $q-1$ voraussetzt. Demgegenüber ist $r_0 > 10^4$ wesentlich schneller zu verifizieren und wird in diesem Zusammenhang als ausreichend angesehen. Für weitere Erläuterungen zu den Bedingungen und Beispielkurven siehe [22].

3.2.b) DSA-Varianten basierend auf Gruppen $E(F_{2^m})$

Um die Systemparameter festzulegen, werden eine elliptische Kurve E und ein Punkt P auf $E(F_{2^m})$ erzeugt, so dass folgende Bedingungen gelten:

- m ist prim.
- $E(F_{2^m})$ ist nicht über F_2 definierbar (d. h. die j -Invariante der Kurve liegt nicht in F_2)
- $\text{ord}(P) = q$ mit q prim.
- $r_0 := \min(r : q \text{ teilt } 2^{mr} - 1)$ ist groß, konkret etwa $r_0 > 10^4$.
- Die Klassenzahl der Hauptordnung, die zum Endomorphismenring von E gehört, ist mindestens 200.

Für den Zeitraum **bis Ende 2009** muss m mindestens 191 betragen. Dabei sollte q sich nur um einen kleinen Faktor von 2^m unterscheiden. Auf jeden Fall ist sicherzustellen, dass die Bitlänge von q mindestens 180 Bit beträgt. **Ab Anfang 2010** muss die Länge von q mindestens 224 Bit betragen, weitere Bedingungen an m werden nicht gestellt.

Die folgende Tabelle fasst die Bitlängen für DSA-Varianten basierend auf Gruppen $E(F_{2^m})$ zusammen.

Parameter \ Zeitraum	bis Ende 2009	bis Ende 2014 ¹
m	191	
q	180	224

Bemerkungen:

- In Bezug auf die oben erwähnten 'Kollisionen' im Sinne von Vaudenay [12] gilt für die auf elliptischen Kurven basierenden Verfahren dasselbe wie für DSA.
- Beim DSA und bei elliptischen Kurven könnte die Wahl bestimmter, ganz spezieller Parameter möglicherweise dazu führen, dass das Verfahren schwächer ist als bei einer zufälligen Wahl der Parameter. Unabhängig davon, wie gravierend man diese Bedrohung einschätzt, kann man dem „Unterschieben“ schwacher Parameter vorbeugend dadurch begegnen, dass bei der Konstruktion der Parameter eine geeignete Einweg-

¹ Siehe Vorbemerkung

funktion, d.h. eine der oben genannten Hashfunktionen, angewandt wird und die Parameter zusammen mit einer nachvollziehbaren entsprechenden Berechnung übergeben werden. Konkrete Vorschläge sind in [1] und [10] zu finden.

4. Erzeugung von Zufallszahlen

Bei der Erzeugung von Systemparametern für Signaturverfahren und für die Schlüsselgenerierung werden Zufallszahlen gebraucht. Bei DSA-ähnlichen Signaturverfahren wird bei jeder Generierung einer Signatur eine neue Zufallszahl benötigt.

Für diese Zwecke bieten sich als Zufallszahlengeneratoren solche Systeme an, die

- eine physikalische Rauschquelle, die beispielsweise auf elektromagnetischen, elektro-mechanischen oder quantenmechanischen Effekten beruht, und
- ggf. eine algorithmische Nachbehandlung der digitalisierten Rauschsignale

besitzen. Die Eigenschaften der digitalisierten Rauschsignalfolge sollten sich hinreichend gut durch ein stochastisches Modell beschreiben lassen. Der physikalische Zufallszahlengenerator sollte ein P2-Generator (Stärke der Mechanismen bzw. Funktionen: hoch) im Sinne der AIS 31 [18] sein; ab Anfang 2011 ist diese Bedingung verbindlich, d. h. der Zufallszahlengenerator *muss* dann ein P2-Generator sein. Qualitativ bedeutet dies: Der durchschnittliche Entropiezuwachs pro Zufallsbit liegt oberhalb einer Mindestschranke. Die Zufallszahlen müssen im laufenden Betrieb statistischen Tests unterzogen werden („Onlinetests“). Der bzw. die Onlinetests sollten dem mathematischen Modell der Rauschquelle angepasst sein. Der bzw. die Onlinetests selbst und das Aufrufschema müssen geeignet sein, nicht akzeptable statistische Defekte oder Verschlechterungen der statistischen Eigenschaften der digitalisierte Rauschsignalfolge in angemessener Zeit zu erkennen. Auf einen Rauschalarm muss angemessen reagiert werden (z. B. weitere Tests, Stilllegen der Rauschquelle). Insbesondere muss ein etwaiger Totalausfall der Rauschquelle umgehend erkannt werden.

Eine aussagekräftige Bewertung eines Zufallszahlengenerators setzt umfassende Erfahrungen voraus. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) verfügt über solche Erfahrungen. Bei Bedarf kann in diesem Zusammenhang auf das Know-how des BSI zurückgegriffen werden.

Als Alternative zu einem physikalischen Zufallszahlengenerator kommt ein Pseudozufallszahlengenerator in Frage. Der innere Zustand des Pseudozufallszahlengenerators wird durch den so genannten Seed initialisiert. In jedem Schritt wird der innere Zustand erneuert und hieraus eine Zufallszahl abgeleitet. Der innere Zustand des Pseudozufallszahlengenerators muss gegen Auslesen und Manipulation (physikalisch, durch Seitenkanalangriffe, über Schnittstelle etc.) ebenso sicher geschützt sein wie die geheimen Signaturschlüssel. Denn mit Kenntnis des inneren Zustands könnte ein potentieller Angreifer zumindest alle zukünftig erzeugten Zufallszahlen mühelos bestimmen.

Für Zertifizierungsdiensteanbieter wird die Verwendung eines physikalischen Zufallszahlengenerators empfohlen. Jeder Pseudozufallszahlengenerator, der im Zusammenhang mit digitalen Signaturen genutzt wird, muss mindestens ein K3-DRNG mit Stärke der Mechanismen bzw. Funktionen „Hoch“ im Sinne der AIS 20 [7] sein. Qualitativ bedeutet dies:

- Es ist einem Angreifer nicht praktisch möglich, zu einer ihm bekannten Zufallszahlenteilfolge Vorgänger oder Nachfolger dieser Teilfolge oder gar einen inneren Zustand zu errechnen, oder diese mit einer Wahrscheinlichkeit zu erraten, die nichtvernachlässigbar über der Ratewahrscheinlichkeit ohne Kenntnis der Teilfolge liegt.

- Die Entropie des Seed beträgt mindestens 80 Bit; empfohlen wird eine Entropie von mindestens 100 Bit.

Anderenfalls muss das entsprechende Verfahren zur digitalen Signatur als potenziell unsicher angesehen werden.

Darüber hinaus sollte der Pseudozufallszahlengenerator ein K4-DRNG mit Stärke der Mechanismen bzw. Funktionen „Hoch“ im Sinne der AIS 20 [7] sein. Qualitativ bedeutet dies, dass zusätzlich folgende Bedingung erfüllt ist:

- Es ist einem Angreifer praktisch unmöglich, aus Kenntnis eines inneren Zustands Vorgängerzufallszahlen oder innere Vorgängerzustände zu errechnen oder diese mit einer Wahrscheinlichkeit zu erraten, die nicht vernachlässigbar über der Ratewahrscheinlichkeit ohne Kenntnis des inneren Zustands liegt.

Die obigen Bedingungen werden **bis Ende 2009** als ausreichend betrachtet. **Ab Anfang 2010** wird darüber hinaus verlangt:

- Die Entropie des Seed beträgt mindestens 100 Bit; empfohlen werden mindestens 120 Bit.
- Der Pseudozufallszahlengenerator muss grundsätzlich der Klasse K4 im Sinne der AIS 20 [7] mit Stärke der Mechanismen bzw. Funktionen "Hoch" angehören. Alternativ genügt eine nachvollziehbare Begründung des Antragstellers, dass das Fehlen der K4-spezifischen Eigenschaft im vorgesehenen Einsatzszenario keine zusätzlichen Sicherheitsrisiken induziert.

Literatur

- [1] NIST: *FIPS Publication 186-2: Digital Signature Standard (DSS)*, Januar 2000 und *Change Notice 1*, Oktober 2001.
- [2] NIST: *FIPS Publication 180-2: Secure Hash Standard (SHS)*, August 2002 und *Change Notice 1*, Februar 2004.
- [3] ISO/IEC 10118-3: *Information technology – Security techniques – Hash functions – Part 3: Dedicated hash functions*, 2nd ed., 2004.
- [4] ISO/IEC 14888-3: *Information technology – Security techniques – Digital signatures with appendix – Part 3: Certificate-based mechanisms*, 1999.
- [5] IEEE P1363: *Standard specification for public key cryptography*, 2000.
- [6] ISO/IEC 9796-3: *Information technology – Security techniques – Digital Signature schemes giving message recovery – Part 3: Discrete logarithm based mechanisms*, 2000.
- [7] AIS 20: *Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren*, Version 1, 2.12.99, samt mathematisch-technischem Anhang (Version 2.0, 2.12.99), <http://www.bsi.bund.de/zertifiz/zert/interpr/aisitsec.htm>
- [8] T. ElGamal: *A public key cryptosystem and a signature scheme based on discrete logarithms*, Crypto '84, LNCS 196, S. 10-18, 1985.
- [9] R. Rivest, A. Shamir, L. Adleman: *A method for obtaining digital signatures and public key cryptosystems*, Communications of the ACM, vol. 21 no. 2, 1978.

- [10] ANSI X9.62-2005: *Public Key Cryptography for the Financial Service Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 2005. (ersetzt ANSI X9.62-1998)
- [11] ISO/IEC 15946-2: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures*, 2002.
- [12] S. Vaudenay: *Hidden collisions in DSS*, Crypto'96, LNCS 1109, S. 83-88, 1996.
- [13] A.K. Lenstra, E.R. Verheul: *Selecting Cryptographic Key Sizes*, J. Cryptology 39, 2001.
- [14] J.-S. Coron, D. Naccache, J. Stern: *On the Security of RSA padding*. Crypto 99, LNCS 1666, 1999.
- [15] PKCS #1 v2.1: RSA Cryptographic Standard, 14.6.2002.
- [16] DIN V66291: *Spezifikation der Schnittstelle zu Chipkarten mit Digitaler Signatur-Anwendung/Funktion nach SigG und SigV, Annex A, 2.1.1*, 1999.
- [17] ANSI X9.31-1998: *Digital signatures using reversible public key cryptography for the financial services industry (rDSA)*, 1998.
- [18] AIS 31: *Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren*, Version 1, 25.9.2001, samt mathematisch-technischem Anhang, (Version 3.1, 25.09.2001),
<http://www.bsi.bund.de/zertifiz/zert/interpr/aisitsec.htm>
- [19] ISO/IEC 15946-4: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 4: Digital signatures giving message recovery*, 2004.
- [20] D. Boneh, G. Durfee: *Cryptanalysis of RSA with private key d less than $N^{0.292}$* . Eurocrypt '99, LNCS 1592, 1999.
- [21] ISO/IEC 9796-2: *Information technology – Security techniques – Digital Signature schemes giving message recovery – Part 2: Integer Factorization based mechanisms*, 2002.
- [22] ECC Brainpool: *ECC Brainpool Standard Curves and Curve Generation*, v. 1.0 (19.10.05), <http://www.ecc-brainpool.org/download/BP-Kurven-aktuell.pdf>; Kurvenparameter als Binärdateien unter <http://www.ecc-brainpool.org/ecc-standard.htm>
- [23] X. Wang, Y. L. Yin, H. Yu: *Collision Search Attacks on SHA-1*, Crypto 2005, LNCS 3621, 2005; Ankündigung signifikanter Verbesserungen bei der "rump session" der Crypto 2005; Vortrag beim NIST Cryptographic Hash Workshop 2005.

Mainz, den 17.12.2007

IS 18

Bundesnetzagentur
für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen
Im Auftrag

D r . W o h l m a c h e r