Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen

Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung

(Übersicht über geeignete Algorithmen)

Vom 30. Dezember 2011

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 17. Juli 2009 (BGBl. I S. 2091), veröffentlicht gemäß Anlage 1 Abschnitt 1 Nr. 2 Signaturverordnung (SigV) vom 16. November 2001 (BGBl. I S. 3074), zuletzt geändert durch die Verordnung vom 15. November 2010 (BGBl. I S. 1542), 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 vom 16. Mai 2001 in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV vom 16. November 2001

Vorbemerkung: Wie in den Vorjahren werden im Folgenden geeignete Algorithmen und Schlüssellängen für den Zeitraum der kommenden sieben Jahre anstatt des in der SigV vorgesehenen Mindestzeitraums von sechs Jahren aufgeführt. Das heißt konkret, dass geeignete Algorithmen und Schlüssellängen bis Ende 2018 statt bis Ende 2017 aufgeführt sind. Im Allgemeinen sind solche längerfristigen Prognosen schwer möglich. Die vorliegende Übersicht über geeignete Algorithmen unterscheidet sich zur zuletzt veröffentlichten Übersicht vom 20. Mai 2011 (BAnz Nr. 85 vom 07. Juni 2011, S. 2034) im Wesentlichen durch das vollständig überarbeitete Kapitel 4.

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 2018) als geeignet anzusehen sind.

Darüber hinaus werden Empfehlungen aufgeführt, die dazu dienen, zukünftigen Entwicklungen im Bereich der kryptographischen Algorithmen und zugehörigen Parameter, die sich heute schon abzeichnen und in Zukunft an Bedeutung zunehmen könnten, zu begegnen. Diese als Empfehlung formulierten Angaben dienen dazu, dem "Interesse der Planungssicherheit der

---

[1] Siehe Vorbemerkung

interessierten Hersteller, Dienstleister und Anwender" Rechnung zu tragen (vgl. Roßnagel/Pordesch: Kommentierung des Signaturgesetzes [35]). Es besteht zum jetzigen Zeitpunkt jedoch keine Pflicht, diese Empfehlungen umzusetzen.

Neben Empfehlungen beinhaltet der Algorithmenkatalog Bemerkungen, die seinem besseren Verständnis dienen und einen rein informativen Charakter haben.

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, Tagungsbände von Konferenzen etc.) und im Internet.

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.

Inhaltsverzeichnis

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 vorgegebener 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 Signaturprüfschlüssel 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. Bei der praktischen Anwendung kommt es jedoch 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) Signaturprüfschlüssel 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

Die beiden Hashfunktionen SHA-1 und RIPEMD-160 sind **bis Ende 2015** nur noch für die Prüfung qualifizierter Zertifikate geeignet.

Folgende Hashfunktionen mit verschiedenen Hashwert-Längen (SHA-256 ist eine 256-Bit Hashfunktion etc.) sind geeignet, ein langfristiges Sicherheitsniveau zu gewährleisten:

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

Diese drei Hashfunktionen sind (mindestens) in den **kommenden sieben Jahren**, d.h. **bis Ende 2018**[1], für die Anwendung bei qualifizierten elektronischen Signaturen geeignet. Die Hashfunktion SHA-224 [2] ist bis **Ende 2015** für die Anwendung bei qualifizierten elektronischen Signaturen geeignet.

Die folgende Tabelle fasst die Eignung der Hashfunktionen zusammen.

geeignet bis Ende **2015**	geeignet bis Ende **2018**
SHA-224, (SHA-1, RIPEMD-160)*	SHA-256, SHA-384, SHA-512

* ausschließlich zur Prüfung qualifizierter Zertifikate, aber nicht zu deren Erstellung oder zur Erzeugung und Prüfung anderer qualifiziert signierter Daten.

Der SHA-1 war bis Ende 2010 zur Erzeugung qualifizierter Zertifikate zugelassen, sofern in die Erzeugung der Seriennummer Zufall mit mindestens 20 Bit Entropie eingeflossen ist. Auch wenn bei der SHA-2-Familie nach gegenwärtigem Kenntnisstand hierfür keine Notwendigkeit besteht, wird dennoch empfohlen, dies auch dort als eine zusätzliche Sicherheitsmaßnahme zu verwenden.

Bemerkung:

- Ob in die Erzeugung eines qualifizierten Zertifikats tatsächlich mindestens 20 Bit Entropie eingeflossen sind, kann im Rahmen der Prüfung des qualifizierten Zertifikats mittels einer Signaturanwendungskomponente gemäß § 2 Nr. 11 b) SigG nicht festgestellt werden. Die Anforderung ist vielmehr vom Zertifizierungsdiensteanbieter in seinem Betrieb zu erfüllen.

3. Geeignete Signaturverfahren

Im Jahr 1977 haben Rivest, Shamir und Adleman das nach ihnen benannte RSA-Verfahren [9] zum Erzeugen und Verifizieren digitaler Signaturen explizit beschrieben. Im Jahr 1984 hat ElGamal [8] ein weiteres Signaturverfahren vorgeschlagen. Eine Variante dieses ElGamal-Verfahrens ist der 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 Punktegruppen *E(K)* elliptischer Kurven über endlichen Körpern *K* basieren, wobei *K* entweder ein endlicher Primkörper F_p oder ein endlicher Körper F_{2^m} der Charakteristik 2 ist.

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

1. RSA-Verfahren [23],
2. DSA [1], [4],
3. DSA-Varianten, basierend auf elliptischen Kurven:
   - EC-DSA [1], [4], [5], [10], [11],
   - EC-KDSA, EC-GDSA [4], [11],
   - Nyberg-Rueppel-Signaturen [6], [19].

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 bzw.
3. dem Diskreten-Logarithmus-Problem in den Gruppen $E(F_p)$ bzw. $E(F_{2^m})$.

Um festzulegen, wie groß die Systemparameter bei diesen Verfahren zu wählen sind, 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], [28]. Solche Prognosen sind nur für relativ kurze Zeiträume sinnvoll (und können sich natürlich jederzeit aufgrund unvorhersehbarer Entwicklungen als falsch erweisen).

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

Die Sicherheit der einzelnen Verfahren ist (mindestens) für die **kommenden sieben Jahre**, d.h. bis **Ende 2018**[1], bei der im Folgenden festgelegten Wahl der Parameter gewährleistet.

3.1. RSA-Verfahren

Der Parameter n muss eine Länge von mindestens 1976 Bit haben. Empfohlen werden 2048 Bit.

Die folgende Tabelle fasst die minimalen Bitlängen zusammen.

Parameter \ Zeitraum	bis Ende **2018**
n	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.

Der öffentliche Exponent *e* wird unabhängig von *n* unter der Nebenbedingung 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 $ed \equiv 1 \mod \text{kgV}(p-1, q-1)$ gilt. Es wird empfohlen, $e \geq 2^{16} + 1$ zu wählen.

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].
- In [3], Table 3-2, ist für 2048-Bit RSA eine Unter- und eine Obergrenze für *e* spezifiziert. ($2^{16} + 1 \leq e < 2^{256}$).

Der Hashwert muss vor der Anwendung des geheimen Exponenten *d* auf die Bitlänge des Moduls formatiert werden. Das Formatisierungsverfahren 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].
- Das Formatierungsverfahren RSA: „Signature Schemes with Appendix“ PKCS#1-v1_5 aus [15] Abschn. 8.2 und 9.2 ist noch bis Ende 2015 geeignet. Für Zertifikatssignaturen ist das PKCS#1-v1_5-Format darüber hinaus bis Ende 2017 geeignet. Es wird aber empfohlen, dieses Verfahren nicht über Ende 2013 hinaus zu verwenden. Der Beweiswert der erstellten Signaturen ist bis Ende 2018 gegeben.

Bemerkung:

- Die Realisierung eines Formatisierungsverfahrens – 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 z.B. [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 der Wert 2^{-100} (siehe [1]; vergleiche aber auch [5] und [17]) empfohlen.

3.2. DSA

Die Bitlänge von *p beträgt* mindestens 2048 Bit. Bis **Ende 2015** muss die Bitlänge des Parameters *q* mindestens 224 Bit betragen. **Ab Anfang 2016** sind für *q* mindestens 256 Bit notwendig. Die folgende Tabelle fasst die Bitlängen für den DSA zusammen.

Parameter \ Zeitraum	bis Ende **2015**	bis Ende **2018**
p	2048	2048
q	224	256

Bemerkungen:

- Zur Erzeugung von *p* und der weiteren Parameter siehe [1]; ab Anfang 2010 soll die Wahrscheinlichkeit, dass *p* oder *q* zusammengesetzt sind, kleiner als 2^{-100} sein.
- In FIPS-186 [1] werden konkrete Werte für die Bitlängen von *p* und *q* vorgegeben.
- Relativ kurze Bitlängen des Parameters *q* erlauben die Konstruktion von 'Kollisionen' im Sinne von Vaudenay [12] bei der Parametergenerierung. Diese Kollisionen haben jedoch in der Praxis der qualifizierten Signaturen keine Bedeutung. Soll dessen ungeachtet die Möglichkeit, diese Kollisionen konstruieren zu können, grundsätzlich ausgeschlossen werden, muss *q* größer als der maximal mögliche Hashwert sein.

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:

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

Für den Parameter *p* gibt es keine Einschränkungen. Die Länge von *q* muss mindestens 224 Bit betragen, und **ab Anfang 2016** sind für *q* mindestens 250 Bit erforderlich.

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

Parameter \ Zeitraum	bis Ende **2015**	bis Ende **2018**
p	keine Einschränkung	keine Einschränkung
q	224	250

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* (mod *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] und [25].

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).
- Es ist $ord(P) = q$ mit *q* prim.
- Für $r_0 := \min(r : q \text{ teilt } 2^{mr}-1)$ gilt $r_0 > 10^4$.
- Die Klassenzahl der Hauptordnung, die zum Endomorphismenring von *E* gehört, ist mindestens 200.

An den Parameter *m* werden keine Bedingungen gestellt. **Ab Anfang 2016** sind für *q* mindestens 250 Bit erforderlich.

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

Parameter \ Zeitraum	bis Ende **2015**	bis Ende **2018**
m	keine Einschränkung	keine Einschränkung
q	224	250

Bemerkungen:

- In Bezug auf die oben erwähnten 'Kollisionen' im Sinne von [12] gilt für die auf elliptischen Kurven basierenden Verfahren dasselbe wie für DSA.
- Ist bei der Berechnung der zweiten Signaturkomponente *s* die Bitlänge des Hashwerts größer als die Bitlänge des Moduls *q*, werden in [10] die überzähligen niederwertigen (rechten) Bits des Hashwerts abgeschnitten. Dies betrifft DSA und DSA-Varianten, die auf Gruppen $E(F_p)$ oder $E(F_{2^m})$ basieren.
- 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 Einwegfunktion, 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], [10], [22] und [25] zu finden.
- Der Leitfaden [30] (Anlage zu [31]) definiert Minimalanforderungen zur Resistenz von Implementierungen elliptischer Kurven über *GF(p)* gegenüber Seitenkanalangriffen.

4. Erzeugung von Zufallszahlen

Bei der Erzeugung von Systemparametern für Signaturverfahren und für die Schlüsselgenerierung werden Zufallszahlen benötigt. Bei DSA-ähnlichen Signaturverfahren wird bei der Generierung jeder Signatur eine neue Zufallszahl benötigt. Für diese Zwecke müssen geeignete Zufallszahlengeneratoren eingesetzt werden.

Die mathematisch-technischen Anlagen [32] und [33] der AIS 20 [7] bzw. der AIS 31 [18] wurden im September 2011 durch die mathematisch-technische Anlage [34] ersetzt. Die für den Algorithmenkatalog relevanten Funktionalitätsklassen wurden (unter neuem Namen) im Wesentlichen beibehalten, und es sind neue Funktionalitätsklassen hinzugekommen, u.a. hybride deterministische Zufallszahlengeneratoren (Funktionalitätsklasse DRG.4) und hybride physikalische Zufallszahlengeneratoren (Funktionalitätsklasse PTG.3).

Neben Zufallszahlengeneratoren, die nach den neuen Evaluierungskriterien evaluiert werden, können auch weiterhin Zufallszahlengeneratoren eingesetzt werden, die nach den alten Evaluierungskriterien evaluiert wurden. Soweit für den Algorithmenkatalog relevant, stellt die nachfolgende Tabelle den neuen Funktionalitätsklassen die entsprechenden alten Funktionalitätsklassen gegenüber. Dies sind keine exakten 1-1-Beziehungen, da die Anforderungen der Funktionalitätsklassen zwar sehr ähnlich, aber nicht identisch sind. Die Anforderungen an die neuen Klassen sind weiter gehend.

neue Funktionalitätsklasse [34]	alte Funktionalitätsklasse [32] bzw. [33]
PTG.2	P2
PTG.3	kein Pendant
DRG.2	K3
DRG.3	K4
DRG.4	kein Pendant

Bemerkungen:

- Hybride Zufallszahlengeneratoren vereinen Sicherheitseigenschaften von deterministischen und physikalischen Zufallszahlengeneratoren.
- Die Sicherheit eines hybriden deterministischen Zufallszahlengenerators der Klasse DRG.4 beruht in erster Linie auf der Komplexität des deterministischen Anteils, welcher der Klasse DRG.3 angehört. Während der Nutzung des Zufallszahlengenerators wird zusätzlich immer wieder neuer Zufall hinzugefügt. Dies kann (z.B.) in regelmäßigen Abständen oder auf die Anforderung einer Applikation hin erfolgen.
- Hybride physikalische Zufallszahlengeneratoren der Klasse PTG.3 besitzen neben einer starken Rauschquelle eine starke kryptographische Nachbearbeitung mit Gedächtnis.
- PTG.3 stellt die stärkste Funktionalitätsklasse in [34] dar.

Grundsätzlich werden physikalische Zufallszahlengeneratoren als geeignet angesehen, die einer der folgenden Klassen angehören:

- Physikalische Zufallszahlengeneratoren
  - neu: PTG.2, PTG.3
  - alt: P2 hoch
- Deterministische Zufallszahlengeneratoren
  - neu: DRG.3, DRG.4
  - alt: K4 hoch mit 100 Bit Seedentropie

Zur Erzeugung von Ephemeralschlüsseln (DSA, ECDSA) wird darüber hinaus empfohlen, einen Zufallszahlengenerator zu wählen, der einer der folgenden Klassen angehört: PTG.3, DRG.3, DRG.4 oder K4 hoch mit 100 Bit Seedentropie (vgl. auch [30]). Hintergrund ist, dass Zufallszahlen, die von PTG.2- oder P2-konformen Zufallszahlengeneratoren erzeugt wurden, z.B. gewisse Schiefen aufweisen können. Es sind gegenwärtig keine Angriffe bekannt, die dies ausnutzen könnten. Vielmehr ist dies eine grundsätzliche Sicherheitsmaßnahme.

Zertifizierungsdiensteanbietern wird empfohlen, einen Zufallszahlengenerator der Klasse PTG.3 zu verwenden.

Zur Erzeugung qualifizierter elektronischer Signaturen (keine Zertifikatssignaturen) kann ein deterministischer Zufallszahlengenerator der Klasse DRG.2 oder K3 hoch verwendet werden,

sofern der Antragssteller nachvollziehbar begründen kann, dass das Fehlen der DRG.3-spezifischen bzw. der K4-spezifischen Eigenschaft (enhanced backward secrecy) in den vorgesehenen Einsatzszenarien keine zusätzlichen Sicherheitsrisiken impliziert.

Sind die Anforderungen an die Zufallszahlengeneratoren nicht erfüllt, muss das entsprechende Verfahren zur qualifizierten elektronischen Signatur als potenziell unsicher angesehen 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.

Die Mindestseedentropie von K3- und K4-Zufallszahlengeneratoren beträgt 100 Bit. Empfohlen werden 120 Bit Seedentropie.

Bemerkungen:

- Die Ableitung von Signaturschlüsseln, Ephemeralschlüsseln und Primzahlen (für RSA) aus den erzeugten Zufallszahlen soll mit geeigneten Algorithmen erfolgen (zu elliptischen Kurven vgl. [30], Abschnitte 5.2 und 5.5.1). Vereinfacht gesagt, sollte einem potentiellen Angreifer so wenig Information über die abgeleiteten (geheim zu haltenden) Werte zur Verfügung stehen wie möglich. Im Idealfall treten alle Werte innerhalb des jeweilig zulässigen Wertebereichs mit derselben Wahrscheinlichkeit auf, und verschiedene Zufallszahlen sollten zumindest keine praktisch ausnutzbaren Zusammenhänge aufweisen.
- Ebenso wie die Signaturalgorithmen kann auch die Erzeugung geheim zu haltender Signaturschlüssel, Ephemeralschlüssel und Primzahlen Ziel von Seitenkanalangriffen werden ([29], [30] etc.). Dieser Aspekt wird [34] explizit angesprochen.

5. Zeitraum und Verfahren zur langfristigen Datensicherung

Damit eine qualifizierte elektronische Signatur auch nach Überschreiten der Eignungsfrist eines Algorithmus, auf dessen Sicherheit die Signatur beruht, ihren Beweiswert erhält und sicher verifiziert werden kann, müssen vor Überschreiten dieser Frist geeignete Maßnahmen nach § 17 SigV getroffen werden. Dazu gehören qualifizierte Zeitstempel, die rechtzeitig vor Überschreiten der Frist erzeugt werden und deren Sicherheit auf längerfristig geeigneten Algorithmen beruht. Vor Überschreiten der Eignungsfrist eines Algorithmus, auf dessen Sicherheit ein solcher qualifizierter Zeitstempel beruht, muss dann dieser wiederum mit einem qualifizierten Zeitstempel längerfristiger Sicherheit versehen werden und so weiter. Die Technische Richtlinie [36] befasst sich mit der langfristigen Beweiswerterhaltung kryptographisch signierter Dokumente.

Statt für jedes einzelne qualifiziert signierte elektronische Datum einen Zeitstempel zu erzeugen, bietet es sich aus Effizienzgründen an, einen einzigen qualifizierten Zeitstempel jeweils für mehrere qualifiziert signierte elektronische Daten zugleich zu erzeugen. Ein geeignetes Verfahren dieser Art ist die Erzeugung so genannter Evidence Records für die qualifizierten elektronischen Signaturen gemäß [24]. Bei der Erzeugung eines solchen Evidence Records wird unter anderem ein Hashbaum erstellt. Für die dafür verwendete Hashfunktion wird hier

sowohl die Kollisionsresistenz als auch die Einwegeigenschaft verlangt. Für die hier einzusetzenden Algorithmen gelten dieselben Eignungsfristen wie zur Erzeugung qualifizierter elektronischer Signaturen.

6. Nicht mehr geeignete kryptographische Algorithmen

In diesem Abschnitt sind alle kryptographischen Algorithmen mit Schlüssellängen und Parametergrößen aufgeführt, die jemals zur Erstellung von qualifizierten elektronischen Signaturen und qualifizierten Zertifikaten geeignet waren und diese Eignung inzwischen aber verloren haben. Diese Algorithmen werden auch weiterhin zur Prüfung von Signaturen oder Zertifikaten benötigt. Dazu müssen diese Algorithmen von den Signaturanwendungskomponenten unterstützt werden.

Die nachfolgenden Tabellen enthalten den letzten Zeitpunkt, an dem der jeweilige Algorithmus mit der angegebenen Schlüssellänge bzw. Parametergröße zur *Erzeugung* qualifizierter elektronischer Signaturen und qualifizierter Zertifikate geeignet war bzw. eine Übergangsfrist endet. (Bei RSA 1024 und SHA-1 wurden Übergangsfristen von 3 bzw. 6 Monaten eingeräumt.) Bei den Hashfunktionen werden zusätzlich die Zeitpunkte angegeben, an denen die Eignung zur *Prüfung* qualifizierter elektronischer Zertifikate erlischt, d.h., bis kurz vor diesem Zeitpunkt ist zur Erhaltung des Beweiswertes für qualifizierte Zertifikate keine Maßnahme nach Kap. 5 notwendig. Für alle anderen qualifiziert signierten Daten sind, falls ihr Beweiswert erhalten bleiben soll, dagegen Maßnahmen durchzuführen.

Hashfunktionen

Hashfunktion	geeignet bis
SHA-1	Ende Juni 2008* Ende 2010** Ende 2015***
RIPEMD-160	Ende 2010 Ende 2015***

* Januar – Juni 2008: Übergangsfrist

** nur noch zur Erzeugung qualifizierter Zertifikate (im Jahr 2010 zusätzlich unter der Auflage von einer Entropie ≥ 20 Bit in der Seriennummer)

*** nur noch zur *Prüfung* qualifizierter Zertifikate

RSA

Modullänge *n*	geeignet bis
756	Ende 2000
1024	Ende März 2008*
1280	Ende 2008
1536	Ende 2009
1728	Ende 2010

* Januar – März 2008: Übergangsfrist

DSA

Parameter *p*	Parameter *q*	geeignet bis
1024	160	Ende 2007
1280	160	Ende 2008
1536	160	Ende 2009
2048	160	Ende 2009

DSA-Varianten basierend auf Gruppen $E(F_p)$

Parameter *p*	Parameter *q*	geeignet bis
keine Einschränkung	160	Ende 2006
192	160	Ende 2005
192	180	Ende 2009

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

Parameter *m*	Parameter *q*	geeignet bis
Keine Einschränkung	160	Ende 2006
191	160	Ende 2005
191	180	Ende 2009

Literatur

[1] NIST: *FIPS Publication 186-3: Digital Signature Standard (DSS),* Juni 2009

[2] NIST: *FIPS Publication 180-3: Secure Hash Standard (SHS)*, Oktober 2008

[3] NIST: *Special Publication 800-78-2: Cryptographic Algorithms and Key Sizes for Personal Identity Verification*, Dezember 2010

[4] ISO/IEC 14888-3: *Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms*, 2006 (ersetzt entsprechende Inhalte von ISO/IEC-14888-3-1998)

[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,* 2006 (ersetzt ISO/IEC 9796-3-2000)

[7] AIS 20: *Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren*, Version 2.0, 19.09.2011, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/ais20_pdf.pdf?__blob=publicationFile

[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. (Zurückgezogen 7.11.2007; ersetzt durch [4] ISO/IEC 14888-3-2006.)

[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 2.0, 19.09.2011,

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/ais31_pdf.pdf?__blob=publicationFile

[19] ISO/IEC 15946-4: *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 4: Digital signatures giving message recovery,* 2004 (zurückgezogen 7.11.2007; ersetzt durch [6] ISO/IEC 9796-3-2006)

[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,* 2010

[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] ISO/IEC 14888-2: *Information technology – Security techniques – Digital signatures with appendix – Part 2: Integer factorization based mechanisms,* 2008 (ersetzt entsprechende Inhalte von ISO/IEC-14888-3-1998)

[24] IETF: *RFC 4998, Evidence Record Syntax (ERS) Standards Track*, August 2007, http://www.ietf.org/rfc/rfc4998.txt

[25] IETF: *RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*, March 2010, http://www.ietf.org/rfc/rfc5639.txt

[26] Gesetz über Rahmenbedingungen für elektronische Signaturen (Signaturgesetz – SigG), http://www.nrca-ds.de

[27] Verordnung zur elektronischen Signatur (Signaturverordnung – SigV), http://www.nrca-ds.de

[28] J.W. Bos, M.E. Kaihara, T. Kleinjung, A.K. Lenstra, P.L. Montgomery: *On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography* (version 2.1, 01.09.2009), http://eprint.iacr.org/2009/389

[29] T. Finke, M. Gebhardt, W. Schindler: *A New Side-Channel Attack on RSA Prime Generation*. CHES 2009, LNCS 5747, 2009

[30] *Minimal Requirements for Evaluating Side-Channel-Attack Resistance of Elliptic Curve Implementations*. Leitfaden,

https://www.bsi.bund.de/DE/Themen/ZertifizierungundAnerkennung/ZertifizierungnachCCundITSEC/AnwendungshinweiseundInterpretationen/AISZertifizierungsschema/aiszertifizierungsschema_node.html

[31] AIS 46: *Informationen zur Evaluierung von kryptographischen Algorithmen und ergänzende Hinweise für die Evaluierung von Zufallszahlengeneratoren.* Version 1 (22.10.2010),

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Zertifizierung/Interpretationen/AIS_46_1_pdf.pdf?__blob=publicationFile

[32] W. Schindler: *Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators*. Version 2.0, 02.12.1999, ehemalige mathematisch-

technische Anlage zur AIS20,

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Zertifierung/Interpretation/AIS20_Functionality_Classes_Evaluation_Methodology_DRNG.pdf?__blob=publicationFile

[33] W. Killmann, W. Schindler: *A proposal for: Functionality classes and evaluation methodology for true (physical) random number generators*. Version 3.1, 25.09.2001, ehemalige mathematisch-technische Anlage zur AIS 31,

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Zertifierung/Interpretation/AIS31_Functionality_classes_evaluation_methodology_for_true_RNG.pdf?__blob=publicationFile

[34] W. Killmann, W. Schindler: *A proposal for: Functionality classes for random number generators*. Version 2.0, 18.09.2011, mathematisch-technische Anlage zur AIS 20 und AIS 31,

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Zertifierung/Interpretation/AIS20_Functionality_classes_for_random_number_generators.pdf?__blob=publicationFile

und

https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Zertifierung/Interpretation/AIS31_Functionality_classes_for_random_number_generators.pdf?__blob=publicationFile

[35] A. Roßnagel (Hrsg.): *Recht der Multimedia-Dienste*, Kommentar zum Informations- und Kommunikationsdienste-Gesetz und Mediendienste-Staatsvertrag, Beck Verlag, München 1999

[36] BSI Technische Richtlinie 03125: *TR-ESOR – Beweiswerterhaltung kryptographisch signierter Dokumente*, Version 1.1, 18.02.2011,

https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03125/BSI_TR_03125_V1.1.pdf?__blob=publicationFile

Mainz, den 30.12.2011

IS 18

Bundesnetzagentur
für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen

Im Auftrag

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