

Geeignete Kryptoalgorithmen gemäß § 17 (2) SigV

Bonn, 29. Mai 1999

Die Sicherheit einer digitalen Signatur hängt primär von der Stärke der zugrundeliegenden Kryptoalgorithmen ab. Im folgenden werden Kryptoalgorithmen genannt, die für digitale Signaturen mindestens für die **kommenden sechs Jahre** (d.h. bis Ende 2004) als geeignet anzusehen sind. Die bit-genauen 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 des vorliegenden Dokuments. Informationen hierzu findet man in der einschlägigen Literatur (Lehrbücher, Proceedings von Konferenzen etc.) und im Internet.

Wir beschränken uns hier auf die wichtigsten praxisrelevanten Algorithmen, deren kryptographische Eigenschaften aufgrund der heute vorliegenden Ergebnisse langjähriger Diskussionen und Analysen am besten eingeschätzt werden können.

Die Zertifizierungsstellen können abweichend von den hier vorgeschlagenen Algorithmen auch andere Verfahren einsetzen, wenn nach Angaben des Bundesamtes für Sicherheit in der Informationstechnik (BSI) deren Eignung von der zuständigen Behörde festgestellt wird.

Die Liste der hier genannten Algorithmen ist als offen und vorläufig anzusehen. Sie wird gemäß der weiteren Entwicklung der kryptologischen Forschung und den Erfahrungen mit praktischen Realisierungen von Signaturverfahren aktualisiert und gegebenenfalls ergänzt werden.

Wichtigste Änderung gegenüber der Fassung vom Januar 1998:

- Bei DSA-Varianten basierend auf elliptischen Kurven E (siehe 3.3.a und 3.3.b) wird für zukünftige Implementierungen gefordert, daß die Klassenzahl der Hauptordnung, die zum Endomorphismenring von E gehört, mindestens 200 ist (zuvor wurde mindestens 100 gefordert).

Wie oben erwähnt betrifft dieses Dokument nur den Zeitraum bis Ende 2004. Es soll aber bereits hier angekündigt werden, daß für den Zeitraum **ab 2005** möglicherweise eine Vergrößerung des Parameters q beim DSA und dessen Varianten sowie des Parameters n beim RSA (siehe Abschnitt 3: *Vorschläge für geeignete Signaturalgorithmen*) notwendig werden könnte.

1. Kryptographische Anforderungen

Ein Schema zur digitalen Signatur im Sinne des Gesetzes umfaßt die folgenden Kryptoalgorithmen:

- Einem Algorithmus zum Hashen von Daten (einer 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,
- einem Signaturalgorithmus, der aus einem Signier- und einem Verifizieralgorithmus besteht. Der Signaturalgorithmus hängt ab von einem Schlüsselpaar, bestehend aus einem

privaten (d.h. geheimen) Schlüssel zum Signieren (Erzeugen einer Signatur) und dem dazugehörigen öffentlichen Schlüssel zum Verifizieren (Prüfen) der Signatur, und

- einem Verfahren zur Erzeugung von Schlüsselpaaren für die einzelnen Teilnehmer.

Ein solches Schema ist sicher, wenn nur der Inhaber des privaten Schlüssels in der Lage ist, eine Signatur zu erzeugen, die vom Verifizieralgorithmus als gültig erkannt wird. (Dies bezieht sich hier nur auf die abstrakt-mathematischen Eigenschaften des Schemas. Sicherheitsprobleme, die bei der konkreten technischen Implementierung auftreten, werden hier ausgeklammert.) Welche Anforderungen sich hieraus an die oben genannten kryptographischen Algorithmen ergeben, soll im folgenden kurz beschrieben werden:

1.1. Hashfunktionen

Beim Signieren wird der Hashwert der zu signierenden Daten gewissermaßen wie ein 'digitaler Fingerabdruck' benutzt. Damit hierbei keine Sicherheitslücke entsteht, muß die Hashfunktion H folgendem Kriterium genügen

- H muß *kollisionsresistent* sein; d. h., es ist praktisch unmöglich, Kollisionen zu finden (Zwei verschiedene digitale Dokumente, die auf denselben Hashwert abgebildet werden, bilden eine Kollision).

Aus der Kollisionsresistenz von H folgt:

- 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. Signaturalgorithmen

Niemand anderes als der Besitzer des Signierschlüssels darf in der Lage sein, Signaturen zu erzeugen. Insbesondere bedeutet dies, daß es praktisch unmöglich ist, den Signierschlüssel aus dem (öffentlichen) Verifizierschlüssel zu berechnen.

1.3 Schlüsselerzeugung

Die verschiedenen Signaturalgorithmen benötigen Schlüssel, die bestimmte Bedingungen erfüllen. In einigen Fällen kommen weitere Bedingungen hinzu, deren Nichtbeachtung zu Schwächen des jeweiligen Verfahrens führen könnte. Im Rahmen dieser Anforderungen müssen die Schlüssel zufällig erzeugt werden (siehe Abschnitt 6.1.5).

2. Vorschläge für geeignete Hashfunktionen

Man kann nach heutigem Kenntnisstand der Analyse von der langfristigen Sicherheit folgender beider 160-Bit Hashfunktionen der MD4-Familie ausgehen:

- RIPEMD-160 ([7], [3]),
- SHA-1 ([2], [3]).

Diese beiden Hashfunktionen sind (mindestens) in den **kommenden sechs Jahren**, d. h. bis Ende 2004, für die Anwendung bei digitalen Signaturen geeignet.

3. Vorschläge für geeignete Signaturalgorithmen

Im Jahr 1977 haben Rivest, Shamir und Adleman als erste ein Verfahren zum Erzeugen digitaler Signaturen explizit beschrieben. Es handelt sich um das nach seinen Erfindern benannte RSA-Verfahren [9]. In 1984 hat El'gamal [8] einen anderen Signaturalgorithmus vorgeschlagen. Eine Variante dieses El'gamal-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 Signaturalgorithmen sind geeignet:

1. RSA [9], [4]
2. DSA [1], [4]
3. DSA-Varianten, basierend auf elliptischen Kurven, und zwar im einzelnen:
 - EC-DSA [11, Abschnitt 8], [4], [5], [10]
 - EC-GDSA (auch „EC-ElGamal“) [11, Abschnitt 6],
 - EC-NR [6], [5],
 - EC-KCDSA [11, Abschnitt 8].

Dies trifft auch für weitere in ISO/IEC 14888-3 beschriebene Verfahren zu. Einige davon werden voraussichtlich in zukünftigen Versionen dieses Dokuments berücksichtigt werden.

Die Sicherheit der oben genannten Verfahren hängt jeweils zusammen mit

1. dem Faktorisierungsproblem für ganze Zahlen,
2. dem Diskreten-Logarithmus-Problem in der multiplikativen Gruppe eines Primkörpern F_p ,
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 gewählt werden müssen, um deren Sicherheit zu gewährleisten, muß man zum einen die besten heute bekannten Algorithmen zum Faktorisieren ganzer Zahlen bzw. zum Berechnen diskreter Logarithmen (in den oben genannten Gruppen) betrachten und zum anderen die Leistungsfähigkeit der heutigen Rechnertechnik berücksichtigen. Um eine Aussage über die Sicherheit für einen bestimmten zukünftigen Zeitraum zu machen, muß man außerdem eine Prognose für die beiden genannten Aspekte zugrunde legen. Solche Prognosen sind nur für relativ kurze Zeiträume möglich (und können sich natürlich jederzeit aufgrund unvorhersehbarer dramatischer Entwicklungen als falsch erweisen).

Die Sicherheit der einzelnen Verfahren ist (mindestens) für die **kommenden sechs Jahre**, d. h. bis Ende 2004, bei folgender Wahl der Parameter gewährleistet:

3.1 RSA

Der zugrunde liegende Modulus $n = pq$ (p und q Primzahlen) soll eine Bitlänge von mindestens 1024 haben:

$$\log_2(n) = \log_2(p) + \log_2(q) > 1023.$$

Für den Zeitraum **bis Ende 2000** reicht eine Minimallänge des Modulus von 768 Bit aus:

$$\log_2(n) > 767.$$

Die Primfaktoren p und q von n sollten ungefähr gleich groß sein, aber nicht zu dicht beieinander liegen, d. h. konkret etwa

$$0,5 < |\log_2(p) - \log_2(q)| < 30$$

Die Primfaktoren p und q werden unter Beachtung der genannten Nebenbedingungen zufällig und unabhängig voneinander erzeugt.

Der öffentliche Exponent e wird mit $ggT(e, (p-1)(q-1)) = 1$ gewählt. Der zugehörige geheime Exponent d wird dann berechnet, so daß $ed \equiv 1 \pmod{(p-1)(q-1)}$ gilt.

Bemerkungen:

1. 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.
2. Der öffentliche Exponent kann zufällig gewählt werden. Auf der anderen Seite haben kleine öffentliche Exponenten den Vorteil, daß die Verifikation der Signatur sehr schnell durchgeführt werden kann. Bei entsprechender Formatierung des Hashwertes, d. h. der Expansion des Hashwertes auf die Blockbreite des asymmetrischen Verfahrens, ist hierbei (im Gegensatz zum Verschlüsseln mit kleinen Exponenten) kein Risiko bekannt.

3.2. DSA

Laut FIPS-186 wird für den Parameter p (p Primzahl) eine Bitlänge von mindestens 512 und höchstens 1024 verlangt. Diese Bedingung wird hier verschärft, 1024 Bit ist die untere Grenze:

$$\log_2(p) > 1023.$$

Zur Erzeugung von p und der weiteren Parameter siehe [1]. Der DSA verlangt für den Parameter q die Bitlänge 160. Dies erlaubt die Konstruktion von 'Kollisionen' im Sinne von [12] bei der Parametergenerierung. Diese Kollisionen haben jedoch in der Praxis keine Bedeutung. Wenn man dessen ungeachtet die Möglichkeit, diese Kollisionen konstruieren zu können, ausschließen möchte, muß man $\log_2(q) \geq 160$ wählen.

3.3 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:

- $\#E(F_p) = a \cdot q$ mit einer von p verschiedenen Primzahl q und

$$\log_2(q) > 159.$$

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

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 aufwendige Faktorisierung von $q-1$ voraussetzt. Demgegenüber ist $r_0 > 10^4$ wesentlich schneller zu verifizieren und wird in diesem Zusammenhang als ausreichend angesehen.

3.3 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:

- $E(F_{2^m})$ ist in keinem echten Teilkörper von F_{2^m} definierbar (d.h. die j -Invariante der Kurve liegt in keinem echten Teilkörper von F_{2^m}).
- $\#E(F_{2^m}) = a \cdot q$ mit q prim und

$$\log_2(q) > 159.$$

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

In Bezug auf die oben erwähnten 'Kollisionen' im Sinne von Vaudenay [12] gilt für die auf elliptische Kurven basierenden Verfahren dasselbe wie für DSA.

Bemerkung: 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 praktische Einwegfunktion, d.h. eine der oben genannten Hash Funktionen angewandt wird und die Parameter zusammen mit einer nachvollziehbaren entsprechenden Berechnung übergeben werden. Konkrete Vorschläge findet man in Appendix 2 and 3 des DSA-Standards [1] und für elliptische Kurven in den Abschnitten F.3.3 und F.3.4 des Entwurfs zum ANSI X9.62-1998 Standards [10].

4. Erzeugung von Zufallszahlen

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

Für diese Zwecke bieten sich als Zufallszahlengeneratoren Systeme an, die

- eine physikalische Rauschquelle, die beispielsweise auf elektromagnetischen, elektro-mechanischen oder quantenmechanischen Effekten beruht, und
- eine kryptographische (bzw. mathematische) Nachbehandlung des Primärausgangs

besitzen. Die Entnahme der Bits aus der physikalischen Rauschquelle sollte sich hinreichend gut durch ein stochastisches Modell beschreiben lassen. Das Primärausgangs sollte, soweit dies technisch möglich ist, permanent einem angepaßten statistischen Test unterzogen werden. Die mathematische Nachbehandlung sollte modellbedingte Abhängigkeiten auflösen.

Zur Schlüsselerzeugung sollte stets ein physikalischer Zufallszahlengenerator verwendet werden.

Wenn für andere Anwendungen (z. B. beim Signieren mit einer DSA-Variante) kein physikalischer Zufallszahlengenerator zur Verfügung steht, kommt als Alternative ein Pseudozufallsgenerator in Frage. Dieser muß durch eine echte Zufallszahl (seed) initialisiert werden. Entscheidend ist, daß eine dem Pseudozufallsgenerator entnommene Bitfolge - wie eine durch physikalischen Zufall erzeugte Bitfolge - die folgende Anforderung erfüllt:

- Es ist a priori keinerlei Aussage über die Bits möglich, die generiert werden. Auch die Kenntnis einer beliebigen Teilfolge erlaubt keinerlei Rückschlüsse über die restlichen Bits.

Jedes Verfahren zur digitalen Signatur wird unsicher, wenn der benutzte Zufallszahlengenerator die genannten Forderungen nicht erfüllt.

Deshalb ist jeder Zufallszahlengenerator auf seine Eignung sorgfältig zu überprüfen. Eine aussagekräftige Bewertung eines Zufallszahlengenerators setzt umfassende Erfahrungen voraus. Das Bundesamt für Sicherheit in der Informationstechnik (BSI) verfügt über solche Erfahrungen, und es wird empfohlen, in diesem Zusammenhang auf das Know-how des BSI zurückzugreifen.

Eine sehr nützliche Zusammenstellung von praktischen Kriterien und Hinweisen für Zufalls- und Pseudozufallsgeneratoren findet man in [5], Abschnitt D6.

Literatur

- [1] NIST: *FIPS Publication 186-1: Digital Signature Standard (DSS)*, Dezember 1998.
- [2] NIST: *FIPS Publication 180-1: Secure Hash Standard (SHS-1)*, Mai 1995.
- [3] ISO/IEC 10118-3: *Information technology – Security techniques - Hash functions - Part 3: Dedicated hash functions*, 1998.
- [4] ISO/IEC 14888-3: *Information technology – Security techniques - Digital signatures with appendix – Part 3: Certificate-based mechanisms*, 1999.

- [5] IEEE P1363/D9: *Standard specification for public key cryptography*, Februar 1999.
- [6] ISO/IEC 9796-4: *Digital signatures schemes giving message recovery – Part 4: Discrete logarithm based mechanisms*, draft, 1999.
- [7] The RIPEMD-160 homepage, www.esat.kuleuven.ac.be/pub/COSIC/bosselae/ripemd/.
- [8] T. El'gamal: *A public key cryptosystem and a signature scheme based on discrete logarithms*, Crypto '84, LNCS, Band 196, S. 10 – 18, Springer-Verlag, 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 X9.62-1998: *Public Key Cryptography for the Financial Service Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, Januar 1999.
- [11] ISO/IEC 15946-2 (WD): *Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 2: Digital signatures*, working draft, 1999.
- [12] Serge Vaudenay: *Hidden collisions in DSS*, Crypto'96, LNCS, Band 1109, S. 83 – 88, Springer Verlag, 1996.