Anhang V zur Vorlesung Kryptologie: Hashfunktionen

von
Peter Hellekalek

Fakultät für Mathematik, Universität Wien, und
Fachbereich Mathematik, Universität Salzburg

Tel: +43-(0)662-8044-5310
Fax: +43-(0)662-8044-137
e-mail: peter.hellekalek@sbg.ac.at
web: http://random.mat.sbg.ac.at/

Salzburg, 29. Juni 2011

Inhaltsverzeichnis

Kapitel 1

Hashfunktionen

Public Key-Kryptographie benötigt für die Erfüllung ihrer Aufgaben bestimmte kryptographische Werkzeuge. Ein besonders wichtiges Werkzeug sind in diesem Zusammenhang *kryptographische Hashfunktionen*. Für Details verweisen wir auf die Monographie von Menezes et al. [MvOV97] und die Lehrbücher von Trappe und Washington [TW06] sowie Stinson [Sti06].

1.1 Einwegfunktionen

1.1 Definition (Einwegfunktion)
Seien X und Y zwei nichtleere Mengen. Eine Funktion $f : X \to Y$ heißt eine *Einwegfunktion*, falls gilt:

1. es gibt effiziente Verfahren zur Berechnung von $f(x)$, $\forall x \in X$,
2. zu gegebenem $y \in Y$ gibt es kein effizientes Verfahren zur Berechnung von passenden Argumenten $x \in X$ mit der Eigenschaft $f(x) = y$.

Anschaulich ausgedrückt spricht man von einer Einwegfunktion, wenn der Wert von $f(x)$ für jedes $x \in X$ leicht zu berechnen ist, hingegen zu gegebenem $y \in Y$ passende $x \in X$ mit $y = f(x)$ aufwändig zu finden sind.

1.2 Bemerkung (Existenzproblem)
Es ist *nicht* bekannt, ob Einwegfunktionen existieren.

1.3 Beispiel (Faktorisierungsproblem)
Sei K eine sehr große natürliche Zahl, zum Bespiel $K = 2^{1024}$. Sei $X = \{(p_1, p_2) : p_1 \neq p_2 \text{ prim }, p_1, p_2 \geq K, i = 1, 2\}$, sei $Y = \mathbb{N}$ und sei

$$\begin{aligned} f: \quad & X \to Y \\ & f(p_1, p_2) = p_1 \cdot p_2 \end{aligned}$$

Es wird vermutet, dass f eine Einwegfunktion ist.

1.4 Beispiel (Diskreter Logarithmus)
Sei p prim, sei $X = \{0, 1, \ldots, p-2\}$, sei $Y = \{1, \ldots, p-1\} = \mathbb{Z}_p^*$ und sei g eine Primitivwurzel modulo p. Dann definieren wir

$$\begin{aligned} f: \quad & X \to Y \\ & f(x) = g^x \pmod{p} \end{aligned}$$

Es wird vermutet, dass f eine Einwegfunktion ist.

1.5 Bemerkung
Für Einwegfunktionen gilt:

1. Bei verschiedenen Anwendungen wie zum Beispiel bei UNIX-Betriebssystemen wird der Paßwort-Klartext mit einer Einwegfunktion (beruht auf einer DES-Variante) verschlüsselt und in einer Systemdatei abgelegt. Beim Einloggen des Benutzers wird der von diesem eingegebene Klartext ebenfalls verschlüsselt und die beiden verschlüsselten Texte werden miteinander verglichen. Auf diese Weise wird das komplizierte Dechiffrieren umgangen.

2. Für die Sicherung von Daten muß eine Einwegfunktion eine effiziente Dechiffrierung erlauben, man verwendet dazu Einwegfunktionen mit Falltüre (E: trap-door one-way functions).

1.6 Definition (Einwegfunktion mit Falltüre)
Unter einer *Einwegfunktion mit Falltüre* (E: trap-door one-way function) verstehen wir eine Einwegfunktion $f : X \to Y$ mit den Eigenschaften

1. es gibt effiziente Verfahren zur Berechnung von $f(x)$, $\forall\ x \in X$, und, für gegebenes $y \in Y$, zur Bestimmung von passenden $x \in X$ mit $f(x) = y$,

2. das effiziente Verfahren zur Berechnung von passenden $x \in X$ mit $f(x) = y$, zu vorgegebenem $y \in Y$, kann nicht aus f hergeleitet werden, sondern benötigt eine (geheim zu haltende) Zusatzinformation, die sogenannte "Falltüre" (E: trap door).

1.7 Bemerkung
Es ist noch nicht gelungen, die Existenz solcher Funktionen zu beweisen.

Einwegfunktionen mit Falltüre lassen sich mit einem Briefkasten mit Schloß vergleichen:

- das Eingeben ist leicht,
- das Herausfinden des Inhalts *ohne* Schlüssel ist schwierig,
- das Herausfinden des Inhalts *mit* Schlüssel ist leicht.

1.2 Hashfunktionen

1.8 Bemerkung (Datenintegrität)
Eine der Grundaufgaben der angewandten Kryptographie lautet, die Integrität einer gegebenen Datei sicherzustellen. Es geht darum feststellen zu können, ob die Datei bei der Speicherung oder der Übermittlung verändert wurde. Diese Aufgabe ist in Zusammenhang mit der digitalen Signatur von grundlegender Bedeutung. Diese Aufgabe wird gelöst, indem man einen elektronischen *Fingerabdruck (E: fingerprint)* der Datei erzeugt, mit Hilfe von kryptographischen Hashfunktionen.

1.9 Definition (Hashfunktion)
Sei A eine nichtleere Menge, Alphabet genannt, und bezeichne A^* die Menge der Wörter über A, also die Menge endlicher Folgen mit Elementen aus A. Unter einer *Hashfunktion* verstehen wir eine Funktion der Gestalt

$$\begin{aligned} h:\ A^* &\to A^n, \\ x &\mapsto h(x), \end{aligned}$$

mit festem $n \in \mathbb{N}$. Wir nennen $h(x)$ den *Hashwert* von x und n die *Länge* der Hashwerte. *Einweg*-Hashfunktionen (E: one-way hash function) sind Hash-Funktionen, die gleichzeitig Einwegfunktionen sind.

1.10 Bemerkung
Eine *Hashfunktion* erzeugt aus einer Eingabe (-Datei) variabler Länge eine Ausgabe (-Datei) fester Länge, den sogenannten Hashwert.

1.11 Bemerkung
Der Hashwert $h(x)$ ist der digitale "Fingerabdruck" der Eingabe (-datei) x. Wenn wir entscheiden sollen, ob zwei Dateien x und x' übereinstimmen, so vergleichen wir deren Hashwerte $h(x)$ und $h(x')$. Hashfunktionen h sind in der Praxis so konstruiert, dass der Fall $h(x) = h(x')$ für $x \neq x'$ extrem unwahrscheinlich ist.

Hash-Funktionen sind "many-to-one", also hochgradig nicht-injektiv. Verschiedene Eingabedateien x und x' können den gleichen Hashwert besitzen.

Bei Einweg-Hashfunktionen ist der Hashwert einer Datei leicht zu berechnen, passende Eingabedateien zu einem vorgegebenen Hashwert sind allerdings wegen der Einweg-Eigenschaft sehr schwer zu finden. "Schwer" bedeutet zur Zeit mehr als 2^{64} Operationen.

Üblicherweise ist der Hashwert wesentlich kürzer als die Eingabedatei.

Moderne Hashfunktionen erzeugen Hashwerte mit einer Länge von 128 Bit oder mehr. Ein wichtiges Konstruktionsziel für Hashfunktionen ist es, dass zu beliebigen Dokumenten $x_1, x_2, \ldots$ die Folge der Hashwerte $h(x_1), h(x_2), \ldots$ nicht von einer Folge von Realisierungen unabhängiger, identisch gleichverteilter Zufallsvariable zu unterscheiden sein soll. Insbesondere versucht man bei der Konstruktion einer Hashfunktion zu erreichen, dass alle möglichen Hashwerte gleich wahrscheinlich sind. Die Wahrscheinlichkeit, dass eine Eingabedatei einen vorgegebenen Hashwert besitzt, ist dann (bei einem 128-Bit Hashwert) 2^{-128}, also verschwindend klein.

Andere Bezeichnungen für kryptographische Hashfunktionen sind

fingerprint	data integrity check (DIC)
message digest	manipulation detection code (MDC)
cryptographic checksum	message authentication code (MAC)
compression function	data authentication code (DAC)
contraction function	

Wir verweisen den interessierten Leser für Details auf Schneier[Sch96].

Es ist naheliegend, dass wir für kryptographische Aufgaben Einweg-Hashfunktionen einsetzen werden. Da nicht bekannt ist, ob es Einwegfunktionen gibt, kann derzeit niemand für eine gegebene Hashfunktion nachweisen, dass es sich um eine Einweg-Hashfunktion handelt. Die folgenden Begriffe spielen bei Sicherheitüberlegungen zu Hashfunktionen eine wichtige Rolle.

1.12 Definition (Kollisionen, Kollisionsresistenz)
Sei die Hash- oder Kompressionsfunktion h gegeben.

Eine *Kollision* zu h ist ein Paar (x, x'), $x \neq x'$, mit der Eigenschaft $h(x) = h(x')$.

Die Funktion h heißt *schwach kollisionsresistent*, wenn es praktisch unmöglich ist, zu gegebenem x ein x', $x' \neq x$, mit $h(x) = h(x')$ zu finden.

Die Funktion h heißt *stark kollisionsresistent*, wenn es praktisch unmöglich ist, eine Kollision (x, x') zu finden.

1.13 Bemerkung (Geburtstagsattacke von Yuval[Yuv79])
Gegeben seien

- das Originaldokument m
- das verfälschte Dokument m'
- die n-Bit Hashfunktion h

Das Konzept der Geburtstagsattacke beruht darauf, dass wir mittels *unauffälliger* Modifikationen an m und m' ein Paar (m_i, m'_j) erhalten, für das gilt:

- gleiche Hashwerte:
$$h(m_i) \;=\; h(m'_j)$$
- m_i ist semantisch gleich m, m'_j ist semantisch gleich m'

Wie wird die Attacke durchgeführt? Dazu geht man in mehreren Schritten vor:

1. Erzeuge $t = 2^{n/2}$ semantisch unauffällige Veränderungen $m_1, m_2, \ldots, m_t$ von m.
2. Berechne die Hashwerte $h(m_1), h(m_2), \ldots, h(m_t)$ und speichere die Paare $(m_i, h(m_i))$, $1 \leq i \leq t$, ab.

3. Erzeuge nacheinander semantisch unauffällige Veränderungen $m'_1, m'_2, \ldots, m'_t$ der verfälschten Nachricht m', berechne jeweils den zugehörigen Hashwert $h(m'_i)$ und vergleiche diesen Hashwert mit den gespeicherten Hashwerten $h(m_1)$, $h(m_2)$, $\ldots$, $h(m_t)$.

4. Sobald wir zwei gleiche Hashwerte gefunden haben, handelt es sich um eine Kollision und wir sind fertig:

$$h(m_i) \;=\; h(m'_j).$$

1.14 Frage
Warum ist diese Attache so gefährlich? Die Nachrichten m_i und m'_j besitzen die gleichen Hashwerte. Es handelt sich bei m'_j um eine Fälschung von m_i, die daher mittels des Hashwertes *nicht* erkannt wird.

1.15 Bemerkung (Praktische Durchführung)
Die praktische Durchführung der Modifikationen kann nach dem folgenden Schema erfolgen: Wir wählen in der Nachricht m eine feste Anzahl (z.B. 80) möglicher Modifikationspunkte, die nichts am Druckbild der Nachricht ändern.

Geeignete Stellen sind zum Beispiel solche, an denen ein zusätzliches Leerzeichen oder ein nicht druckbares Zeichen nicht auffallen oder an denen man ein Leerzeichen durch ein Tabulatorzeichen ersetzen kann. Wir können dann an diesen Stellen die genannten Veränderungen durchführen und auf diese Weise die vielen semantisch identen Variationen der Nachricht m erzeugen, die wir bei der Geburtstagsattacke benötigen.

1.16 Frage
Hat die Geburtstagsattacke von Yuval eine Aussicht auf Erfolg? Bei der Beantwortung dieser Frage werden wir sehen, dass die Wahl $t = \sqrt{2^n}$ Sinn macht.

1.17 Bemerkung (Fragen)
Wir erzeugen mit dieser "Fälschungstechnik" sehr viele verschiedene Nachrichten und damit sehr viele Hashwerte.

Die folgenden Fragen stellen sich:

Frage 1 Wie groß ist die Wahrscheinlichkeit, dass sich unter q zufällig erzeugten Hashwerten $y_1, \ldots, y_q$ zwei gleiche befinden?

Frage 2 Sei y ein gegebener Hashwert. Wie wahrscheinlich ist es, dass sich unter q zufällig erzeugten Hashwerten $y_1, \ldots, y_q$ ein Hashwert y_j befindet mit $y_i = y$?

Die Antwort auf diese Fragen ist seit langem bekannt und Gegenstand der Beispielsammlung zur stochastischen Modellbildung: das Stichwort lautet "Geburtstagsparadoxon" oder "Geburtstagsproblem".

1.3 Zum Geburtstagsproblem

1.18 Bemerkung (Ein Urnenproblem)
Eine Urne enthalte r Kugeln, die von 1 bis r numeriert seien. Wir ziehen nach-

einander mit Zurücklegen insgesamt q Kugeln aus der Urne.

Frage: Wie groß ist die Wahrscheinlichkeit $\beta(r,q)$, dass sich unter den q gezogenen Kugeln mindestens zwei mit der gleichen Nummer befinden?

1.19 Satz
Es gilt

$$\beta(r,q) \approx (\geq) 1 - \exp\left(-\frac{q\cdot(q-1)}{2\cdot r}\right). \tag{1.1}$$

Beweis.
Es gilt offensichtlich

$$\beta(r,q) = 1 - \left(1\cdot\left(1-\frac{1}{r}\right)\cdot\ldots\cdot\left(1-\frac{q-1}{r}\right)\right). \tag{1.2}$$

Für kleine x gilt die Approximation

$$1 - x \approx (\leq) \exp(-x), \tag{1.3}$$

wobei der Fehler kleiner als $x^2/2$ ist. Dies folgt aus der Potenzreihenentwicklung der Exponentialfunktion $\exp(x)$:

$$\exp(-x) = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} \pm \cdots,$$

Daraus schließen wir

$$\begin{aligned}
&\left(1-\frac{1}{r}\right)\cdot\left(1-\frac{2}{r}\right)\cdot\ldots\cdot\left(1-\frac{q-1}{r}\right)\\
\approx (\leq)\ &\left(\exp\left(-\frac{1}{r}\right)\right)\cdot\left(\exp\left(-\frac{1}{r}\right)\right)^2\cdots\left(\exp\left(-\frac{1}{r}\right)\right)^{q-1}\\
=\ &\exp\left(-\frac{1+2+\cdots+q-1}{r}\right).
\end{aligned}$$

□

1.20 Korollar (Anzahl der Versuche)
Eine Urne enthalte r Kugeln, die von 1 bis r numeriert seien. Wir geben eine Wahrscheinlichkeit $\beta(r,q)$ vor und fragen nach der Stichprobengröße q, bei der mit einer Wahrscheinlichkeit mindestens gleich $\beta(r,q)$ eine Kollision in der Stichprobe vorhanden ist (d.h. in der Stichprobe befinden sich zwei gleiche Kugeln).

Dazu lösen wir die Näherungsgleichung (1.1) nach q auf. Es gilt dann

$$\begin{aligned}
\exp\left(-\frac{q\cdot(q-1)}{2\cdot r}\right) &\approx (\geq) \quad 1-\beta(r,q)\\
\Longrightarrow \qquad -\frac{q\cdot(q-1)}{2\cdot r} &\approx (\geq) \quad \ln(1-\beta(r,q)),\\
\Longrightarrow\ q(q-1) = q^2 - q &\approx (\leq) \quad 2\cdot r\cdot\ln\frac{1}{1-\beta(r,q)}.
\end{aligned}$$

Für große Werte von q können wir den Term $-q$ vernachlässigen. Dies ist zum Beispiel bei Hashfunktionen der Fall. Wir erhalten

$$q \approx \sqrt{2 \cdot r \cdot \ln \frac{1}{1-\beta(r,q)}}. \tag{1.4}$$

1.21 Bemerkung (Klassisches Geburtstagsparadoxon)
Sei $r = 365$ und sei $\beta(r, q) = 0.5$ gewählt. Die Zahl $r = 365$ steht also für die Anzahl der möglichen Geburtstage. Dann folgt aus der Approximation (1.4) die Aussage $q \approx 22.49$. Mit einer Wahrscheinlichkeit von mindestens 0.5 finden sich in einer zufällig ausgewählten Gruppe von 23 Personen zwei Personen mit dem gleichen Geburtstag.

1.22 Bemerkung (Starke Kollision: Stichprobengröße)
Sei wiederum $\beta(r, q) = 0.5$ gewählt und bezeichne r die Anzahl der möglichen Hashwerte.

Aus Approximation (1.4) folgt die Beziehung

$$q \approx 1.17 \cdot \sqrt{r}$$

Mit anderen Worten, mit Wahrscheinlichkeit 0.5 kommt in einer Stichprobe von ungefähr $\sqrt{r}$ zufälligen Hashwerten eine Kollision vor.

Diese Tatsache ist der Grund dafür, dass man heute Hashfunktionen mit mindestens 128 Bit Hashlänge oder mehr verwendet.

1.23 Bemerkung (Schwache Kollision)
Bezeichne r die Anzahl der möglichen Hashwerte und sei ein Dokument x mit Hashwert y gegeben.

Wie groß ist die Wahrscheinlichkeit $\gamma(r, q)$, dass unter q zufällig gewählten Dokumenten ein Dokument x' den gleichen Hashwert y wie das Dokument x besitzt?

Wir leiten aus unserem Urnenmodell das folgende Resultat ab.

1.24 Lemma
Es gilt

$$\gamma(r, q) \approx (\geq)\ 1 - \exp\left(-\frac{q}{r}\right).$$

Beweis.
Sei y der Hashwert des gegebenen Dokumentes x und sei x' ein zufällig gewähltes Dokument. Dann gilt

$$\mathbf{Pr}(\text{Hashwert von } x' \text{ ist verschieden von } y) = \frac{r-1}{r}.$$

Daraus folgt sofort für q zufällig gewählte Dokumente

$$\mathbf{Pr}(\text{für } q \text{ Dokumente sind deren Hashwerte} \neq y) = \left(\frac{r-1}{r}\right)^q$$

Unter Beachtung der Approximation (1.3) ergibt sich

$$\gamma(r,q) \;=\; 1-\left(1-\frac{1}{r}\right)^q \;\approx(\geq)\; 1-\exp\left(-\frac{q}{r}\right). \tag{1.5}$$

□

1.25 Korollar (Anzahl der Versuche)
Sei y der Hashwert des gegebenen Dokumentes x und sei die Kollisionswahrscheinlichkeit $\gamma(r,q)$ gegeben. Wir erzeugen q zufällige Dokumente. Wie müssen wir die Stichprobengröße q wählen, damit unter den q zufälligen Dokumenten mindestens ein Dokument den Hashwert y besitzt?

Wir lösen dazu die Beziehung (1.5) nach q auf:

$$\begin{aligned}
\exp\left(-\frac{q}{r}\right) &\approx(\geq) \quad 1-\gamma(r,q)\\
-\frac{q}{r} &\approx(\geq) \quad \ln(1-\gamma(r,q))\\
q &\approx(\leq) \quad r\cdot\ln\frac{1}{1-\gamma(r,q)}.
\end{aligned} \tag{1.6}$$

1.26 Bemerkung (Klassisches Geburtstagsparadoxon)
Sei $r=365$ und sei $\gamma(r,q)=0.5$ gewählt. Sei weiters ein bestimmter Geburtstag y vorgegeben. Die Zahl $r=365$ steht wieder für die Anzahl der möglichen Geburtstage. Dann folgt aus der Approximation (1.6) die Aussage $q\approx 253$. Mit einer Wahrscheinlichkeit von mindestens 0.5 finden sich in einer zufällig ausgewählten Gruppe von 253 Personen eine Person mit dem vorgegebenen Geburtstag y.

1.27 Bemerkung (Schwache Kollision: Anzahl der Versuche)
Sei wiederum $\gamma(r,q)=1/2$ gewählt und bezeichne r die Anzahl der möglichen Hashwerte.

Aus Approximation (1.6) folgt die Beziehung

$$q\approx 0.69\cdot r$$

Mit anderen Worten, mit Wahrscheinlichkeit 0.5 kommt in einer Stichprobe von ungefähr $0.69\cdot r$ zufälligen Hashwerten eine Kollision mit einem vorgegebenen Hashwert y vor.

1.28 Bemerkung (Schutz vor der Geburtstagsattacke)
Wie schützen Sie sich vor der Geburtstagsattacke von Yuval? Dies ist recht einfach: bevor Sie ein Dokument digital signieren, verändern Sie es unwesentlich, etwa durch Einfügen eines Leerzeichens. Sie erhalten auf diese Weise ein Dokument m''. Mit großer Wahrscheinlichkeit ändert sich damit der Hashwert und es wird somit gelten: $h(m'')\neq h(m'_j)$.

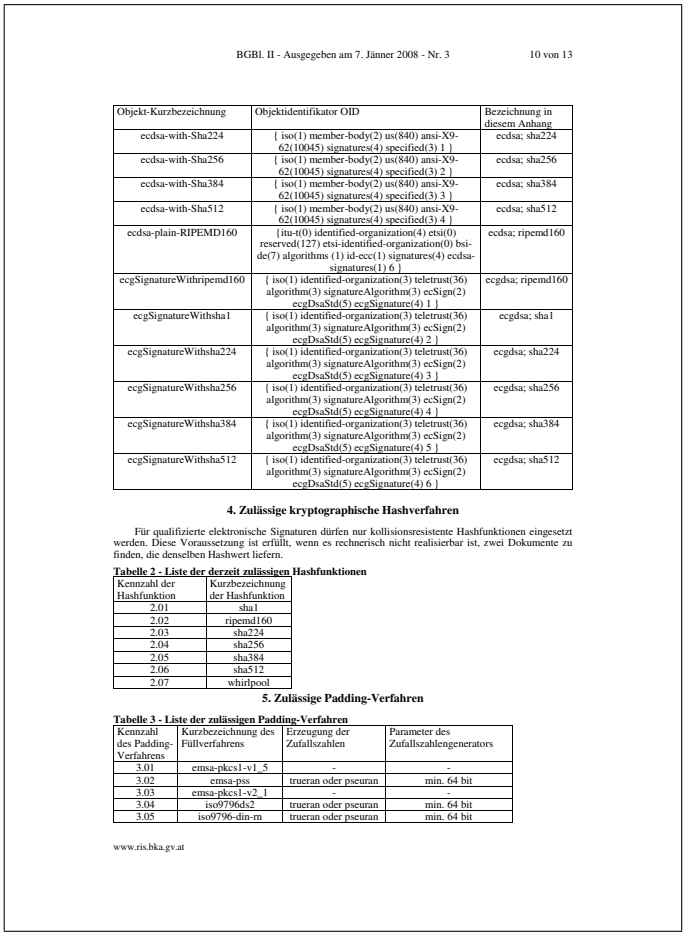
Somit: bevor Sie ein Dokument digital signieren, das Ihnen vorgelegt wird, verändern Sie es und signieren Sie das veränderte Dokument.

1.4 Links

Online-Material zu zahlreichen kryptographischen Themen ist verfügbar unter `http://webcourse.cs.technion.ac.il/236506/Winter2008-2009/en/lnk.html`

Die Folien zur Vorlesung über Kryptologie von Eli Biham sind unter `http://webcourse.cs.technion.ac.il/236506/Winter2008-2009/en/ho_Lectures.html` verfügbar.

Die Signaturverordnung sowie das Signaturgesetz finden Sie bei der Rundfunk- und Telekom Regulierungs-GmbH unter `http://www.rtr.at/`, siehe auch Abbildung 1.1.



BGBl. II - Ausgegeben am 7. Jänner 2008 - Nr. 3 10 von 13

Objekt-Kurzbezeichnung	Objektidentifikator OID	Bezeichnung in diesem Anhang
ecdsa-with-Sha224	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) specified(3) 1 }	ecdsa; sha224
ecdsa-with-Sha256	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) specified(3) 2 }	ecdsa; sha256
ecdsa-with-Sha384	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) specified(3) 3 }	ecdsa; sha384
ecdsa-with-Sha512	{ iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) specified(3) 4 }	ecdsa; sha512
ecdsa-plain-RIPEMD160	{itu-t(0) identified-organization(4) etsi(0) reserved(127) etsi-identified-organization(0) bsi-de(7) algorithms (1) id-ecc(1) signatures(4) ecdsa-signatures(1) 6 }	ecdsa; ripemd160
ecgSignatureWithripemd160	{ iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecgDsaStd(5) ecgSignature(4) 1 }	ecgdsa; ripemd160
ecgSignatureWithsha1	{ iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecgDsaStd(5) ecgSignature(4) 2 }	ecgdsa; sha1
ecgSignatureWithsha224	{ iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecgDsaStd(5) ecgSignature(4) 3 }	ecgdsa; sha224
ecgSignatureWithsha256	{ iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecgDsaStd(5) ecgSignature(4) 4 }	ecgdsa; sha256
ecgSignatureWithsha384	{ iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecgDsaStd(5) ecgSignature(4) 5 }	ecgdsa; sha384
ecgSignatureWithsha512	{ iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecgDsaStd(5) ecgSignature(4) 6 }	ecgdsa; sha512

4. Zulässige kryptographische Hashverfahren

Für qualifizierte elektronische Signaturen dürfen nur kollisionsresistente Hashfunktionen eingesetzt werden. Diese Voraussetzung ist erfüllt, wenn es rechnerisch nicht realisierbar ist, zwei Dokumente zu finden, die denselben Hashwert liefern.

Tabelle 2 - Liste der derzeit zulässigen Hashfunktionen

Kennzahl der Hashfunktion	Kurzbezeichnung der Hashfunktion
2.01	sha1
2.02	ripemd160
2.03	sha224
2.04	sha256
2.05	sha384
2.06	sha512
2.07	whirlpool

5. Zulässige Padding-Verfahren

Tabelle 3 - Liste der zulässigen Padding-Verfahren

Kennzahl des Padding-Verfahrens	Kurzbezeichnung des Füllverfahrens	Erzeugung der Zufallszahlen	Parameter des Zufallszahlengenerators
3.01	emsa-pkcs1-v1_5	-	-
3.02	emsa-pss	trueran oder pseuran	min. 64 bit
3.03	emsa-pkcs1-v2_1	-	-
3.04	iso9796ds2	trueran oder pseuran	min. 64 bit
3.05	iso9796-din-rn	trueran oder pseuran	min. 64 bit

www.ris.bka.gv.at

Abbildung 1.1: Ausschnitt aus der Signaturverordnung 2008

Literaturverzeichnis

[MvOV97] A. J. Menezes, P. C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.

[Sch96] B. Schneier. *Applied Cryptography*. Wiley, New York, second edition, 1996.

[Sti06] D. R. Stinson. *Cryptography*. Chapman and Hall/CRC Press, Boca Raton, 3rd edition, 2006.

[TW06] W. Trappe and L. Washington. *Introduction to Cryptography with Coding Theory*. Pearson Prentice Hall, 2nd edition, 2006.

[Yuv79] Gideon Yuval. How to Swindle Rabin. *Cryptologia*, **3**(3):187–189, 1979.