



---

## Aufsichtsstelle für elektronische Signaturen

Telekom-Control-Kommission  
und  
Telekom-Control  
Österreichische Gesellschaft für  
Telekommunikationsregulierung mbH

### Konsultation

zu den Anforderungen des Signaturgesetzes  
an die Geräte der Benutzer

## Einleitung

Das Signaturgesetz schafft neue Möglichkeiten für die elektronische Kommunikation in Österreich. Mit der rechtlichen Anerkennung der elektronischen Signatur ist die wesentlichste Voraussetzung für ein raschen Übergang zur Anwendung sicherer elektronischer Kommunikationsformen geschaffen worden.

Der Gesetzgeber hat sich bewusst dafür entschieden, das Signaturgesetz technologie-neutral zu gestalten. Das Gesetz soll breiten Raum für die verschiedensten Signaturverfahren gewähren und alle Verfahren durch einen gemeinsamen Rechtsrahmen absichern.

Die Vielfalt an Verfahren, Protokollen und Formaten in elektronischen Netzen, allen voran im Internet, schafft aber auch ein großes Maß an Unklarheit über die neuen technischen Möglichkeiten, die durch sie geschaffene Sicherheit und die damit verbundenen Gefahren.

Ein Problemkreis, aus dem immer wieder Fragen an die Aufsichtsstelle herangetragen wurden, betrifft die Sicherheitsmaßnahmen, die auf den Geräten der Benutzer vorgenommen werden können bzw. müssen.

## Zielsetzung

Mit dieser Konsultation möchte die Telekom-Control-Kommission als Aufsichtsstelle für elektronische Signaturen mehrere Ziele erreichen:

- Das Signaturgesetz bezieht sich an der für Sicherheitsfragen entscheidenden Stelle (§ 18 Abs. 5 SigG) auf den „Stand der Technik“. Mit der Konsultation soll der derzeitige Stand der Technik erkundet werden, um festzustellen, wie die Anforderungen des Signaturgesetzes und der Signaturverordnung zurzeit umgesetzt werden können. Die das Signaturgesetz ausgestaltende Signaturverordnung liegt als Entwurf vor. Die bereits beschlossene endgültige Fassung der Signaturverordnung wird voraussichtlich Ende Jänner im Bundesgesetzblatt kundgemacht werden. In den im gegenständlichen Zusammenhang relevanten Punkten sind aber keine wesentlichen Änderungen zu erwarten.
- Eine wesentliche Fragestellung bei allen Sicherheitsproblemen ist die Verteilung der Verantwortung auf die verschiedenen Akteure. Für welche Aspekte der Sicherheit die Anbieter die Verantwortung übernehmen können und welche Aspekte im Verantwortungsbereich der jeweiligen Benutzer liegen, hängt stark von den technischen Möglichkeiten ab, die den Anbietern und Benutzern zur Verfügung stehen. Die Konsultation soll auch zur Klärung dieser Fragestellungen beitragen.
- Welche Technologien zurzeit verfügbar sind, ist nicht nur der Aufsichtsstelle und den Anbietern, sondern vor allem auch den potenziellen Nutzern nicht vollständig bekannt. Das Ergebnis des Konsultationsverfahrens soll auch diesbezüglich mehr Klarheit schaffen.

Die Aufsichtsstelle lädt alle Interessierten, insbesondere aber Anbieter von Zertifizierungsdiensten und alle potenziellen Nutzer sicherer elektronischer Signaturverfahren – vor allem Behörden und E-Commerce-Anbieter – dazu ein, zur Konsultation Stellung zu nehmen. Um Übersendung der Stellungnahmen in elektronisch weiterverarbeitbaren Formaten (bevorzugt Word 97) an die E-Mail-Adresse [signatur@tkc.at](mailto:signatur@tkc.at) wird ersucht.

Die Aufsichtsstelle wird die eingesandten Stellungnahmen auf der Website <http://www.tkc.at> veröffentlichen und geht davon aus, dass jeder Einsender sich durch die Übermittlung der

Stellungnahme mit der Veröffentlichung einverstanden erklärt. Die Aufsichtsstelle behält sich vor, Stellungnahmen zu kürzen oder gar nicht zu veröffentlichen. Ein Rechtsanspruch auf Veröffentlichung besteht nicht. Die Ergebnisse des Konsultationsverfahrens werden in zusammengefasster Form veröffentlicht. Einsendeschluss ist der **20.02.2000**.

Im Hinblick auf die Zielsetzung der Konsultation lädt die Aufsichtsstelle ausdrücklich auch kommerzielle Anbieter von Sicherheitsprodukten dazu ein, im Konsultationsverfahren Stellung zu nehmen und ihre Produkte darzustellen. Es wird aber darauf hingewiesen, dass reine Werbung nicht erwünscht ist und die Aufsichtsstelle unsachliche Stellungnahmen nicht berücksichtigen wird.

Da mit der Konsultation der derzeitige Stand der Technik erkundet werden soll, wird in der Konsultation versucht, die derzeit üblichen technischen Begriffe zu verwenden – im Unterschied zur Terminologie des Signaturgesetzes, bei dem versucht wurde, möglichst technologieneutrale Begriffe zu verwenden, um auch zukünftige Technologien zu berücksichtigen. Wir sprechen also von „privaten Schlüsseln“ statt von „Signaturerstellungsdaten“ und von „öffentlichen Schlüsseln“ statt von „Signaturprüfdaten“. Wenn vom „Anbieter“ die Rede ist, ist in der Regel ein „Zertifizierungsdiensteanbieter“ im Sinne des Signaturgesetzes gemeint, der qualifizierte Zertifikate und sichere elektronische Signaturverfahren anbietet. Um die Problemstellungen anschaulich zu machen, haben wir auch weit verbreitete Betriebssysteme und Anwendungsprogramme wie z. B. Windows 95 und Microsoft Word namentlich angesprochen. Damit soll weder eine Bevorzugung bestimmter Software zum Ausdruck kommen, noch wollen wir den Eindruck erwecken, diese Software würde besondere Unsicherheiten aufweisen. Es soll lediglich vermieden werden, dass Problemstellungen unklar bleiben, weil sie in zu allgemeiner Form dargestellt werden.

## **Grundsätzliche Sicherheitsprobleme**

Die Aufsichtsstelle hat festgestellt, dass im Zusammenhang mit den Anforderungen an die Geräte der Benutzer vor allem vier große Themenkreise problematisiert wurden:

**Dokumentenformate.** Bei einem elektronisch signierten Dokument soll sich unbeschränkt lange Zeit in einfachster Weise feststellen lassen, was eigentlich signiert wurde. Das Dokument soll also lesbar bleiben und darf sich nicht nachträglich verändern lassen oder von selbst verändern. Das Dokument soll außerdem auf verschiedenen Rechnern mit verschiedenen Betriebssystemen immer in gleicher Weise angezeigt werden. Dieses Ziel steht im Widerstreit zum Ziel, mit einer immer größeren Anzahl an Dokumentenformaten einem ständig wachsenden Bedarf an Funktionalität gerecht zu werden.

**Speicherung der privaten Schlüssel.** Derzeit ist es üblich, dass private Schlüssel in einer Datei auf der Festplatte des Benutzers (in der Regel verschlüsselt durch ein Passwort, das der Benutzer frei wählen kann) abgespeichert werden. Das Signaturgesetz verlangt jedoch, dass die „unbefugte Verwendung von Signaturerstellungsdaten verlässlich verhindert“ ist (§ 18 Abs. 1 SigG). Diese starke Forderung in Richtung erhöhter Sicherheit bewirkt einen Paradigmenwechsel, wobei die derzeit üblichen Softwarelösungen durch Hardwarelösungen (zu Beginn wird es sich in der Regel um Chipkarten handeln) ersetzt werden.

**Kontrolle über den Signaturvorgang.** Die eben genannte Forderung des Gesetzes, die unbefugte Verwendung von privaten Schlüsseln verlässlich zu verhindern, hat auch starke Implikationen auf die Anforderungen an die gesamte auf dem Rechner des Signators laufende Software. Wer keine Kontrolle über sein Betriebssystem und seine Anwendungsprogramme hat, kann letztlich auch nicht mehr steuern, was durch die Chipkarte signiert wird, die er in seinen Kartenleser steckt.

**Verwendung von Schlüsseln für andere Zwecke.** Die eingesetzten kryptographischen Verfahren lassen sich nicht nur für elektronische Signatur, sondern auch für Verschlüsselung, Authentifizierung und Finanztransaktionen einsetzen. Typischerweise wird der Anbieter von Signatursoftware daher auch andere Sicherheitsfunktionen in seinem Produkt anbieten. Für den Benutzer muss jedoch klar sein, ob er mit der Eingabe eines PIN-Codes ein Dokument entschlüsselt oder einen Vertrag mit weit reichenden rechtlichen Folgen abschließt. Wie man die verschiedenen Einsatzmöglichkeiten von Kryptographie in einer für den Benutzer transparenten Weise trennen kann, ist in vielerlei Hinsicht noch unklar.

In allen anzusprechenden Fragen wird sich als Hauptproblem immer wieder zeigen, dass Sicherheit und Funktionalität einander grundsätzlich widersprechen. Je höher man die Anforderungen an die Sicherheit legt, desto größer sind die Einschränkungen für die Benutzer. Stellt man zu hohe Anforderungen an die Sicherheit, dann werden die Benutzer diese ignorieren oder umgehen. Will man ein Maximum an Funktionalität schaffen, dann beeinträchtigt man die Sicherheit. Gerade jene Benutzer, die aufgrund ihrer Neugier und Bereitschaft, sich mit neuen Technologien auseinanderzusetzen, besonderes Interesse an den neuen Möglichkeiten elektronischer Signatur haben, werden daher feststellen müssen, dass sich hohe Sicherheit nur dann erreichen lässt, wenn man zu Lasten der Funktionalität darauf verzichtet, jeweils die neuesten Betriebssysteme und Anwendungsprogramme zu verwenden, deren Auswirkungen auf die Sicherheit noch wenig erforscht sind.

# 1. Dokumentenformate

## Anforderungen

Die Anforderungen, die man an ein elektronisch signiertes Dokument stellt, können relativ klar zusammengefasst werden:

- Das Dokument soll lesbar sein – auch in ferner Zukunft.
- Das Dokument soll nachträglich nicht verändert werden können und sich nicht selbst verändern können, ohne dass man das bemerkt.
- Es muss klar sein, was signiert wurde und was nicht. Es soll keine Bestandteile geben, die auf einem PC unsichtbar sind und auf einem anderen sichtbar.

Diese klaren und auf den ersten Blick völlig selbstverständlichen Forderungen werden über weite Strecken von den bestehenden Anwendungsprogrammen erfüllt. In der Regel lässt sich ein mit einem weit verbreiteten Textverarbeitungsprogramm geschriebener Text dank entsprechender Filterprogramme auch noch nach vielen Jahren mit anderen Textverarbeitungsprogrammen öffnen und anzeigen. In der Regel kann man dynamische Bestandteile eines Textes (z. B. das häufig verwendete Feld „aktuelles Datum“) von statischen Bestandteilen leicht unterscheiden. In der Regel wird ein Dokument auf allen Betriebssystemen und allen Anwendungsprogrammen halbwegs gleich dargestellt.

Die oben gestellten Anforderungen hundertprozentig umzusetzen ist aber angesichts der explodierenden Anzahl von Datenformaten praktisch unmöglich. Durch die Konsultation sollen die beim derzeitigen Stand der Technik bestehenden Möglichkeiten aufgezeigt werden, den Anforderungen der im Entwurf vorliegenden Signaturverordnung – insbesondere des § 7 SigV – zu entsprechen.

## Übersicht über gängige Datenformate

### Ascii (text/plain)

Das verhältnismäßig unproblematischste Dokumentenformat ist das reine Textformat Ascii (text/plain). Es kommt ohne jegliche Formatierung aus. Ein Programm, das eine Ascii-Datei anzeigen soll, stellt einfach jedes einzelne Zeichen der Datei der Reihe nach am Bildschirm dar.

Auch bei diesem simplen Format gibt es einige Unklarheiten. Je nach Betriebssystem ist z. B. das Zeilenende mit dem Code 10 (Unix), mit dem Code 13 (MacIntosh) oder mit der Kombination der Codes 13 und 10 (DOS, Windows, viele Internetformate) kodiert. Schwierigkeiten bereiten auch die Sonderzeichen, die in dem ursprünglich amerikanischen Format nicht vorgesehen waren und nun in zahlreichen Varianten in unterschiedlicher Weise codiert werden. Im Prinzip lassen sich Ascii-Dateien und ihre Varianten aber auf fast jedem Rechner der Welt darstellen, lesen und ausdrucken.

### HTML (text/html)

Das beliebteste Format im Internet bot ursprünglich eine einfache und überschaubare Methode an, Texte zu formatieren. Das Format basiert auf dem Ascii-Format und ist damit auch ohne spezielle Software für (geübte) menschliche Augen lesbar. Von einem Browser werden aber jene Stellen, die in spitze Klammern gesetzt sind, als Formatierungsanweisungen interpretiert. Eine Stelle, die bei Interpretation der Datei als

Ascii-Text z. B. so aussehen würde: Elektronische *<i>Signaturen</i>* sind **<b>sicher</b>**., interpretiert ein Browser, indem er das Wort „Signaturen“ kursiv (italics) und das Wort „sicher“ im Fettdruck (bold) darstellt.

Die besondere Stärke (aber auch eine besondere Schwäche) von HTML liegt darin, dass das Format für Weiterentwicklungen offen ist. Wenn ein Browser auf eine Anweisung in spitzen Klammern stößt, die er nicht interpretieren kann, so soll er sie einfach ignorieren. Auf diese Weise können auch alte Browser, die für die HTML-Version 1.0 geschrieben wurden, Texte im neuesten HTML 4.0 darstellen.

Diese Offenheit für Erweiterungen hat aber auch dazu geführt, dass die Marktführer Microsoft und Netscape in ihrem „Browserkrieg“ ständig neue Funktionalitäten erfanden. Das World Wide Web Consortium konnte diese konkurrierenden Entwicklungen nur mühsam zu einheitlichen Standards zusammenfassen. In der Realität existiert heute kaum eine HTML-Datei, die auch nur irgendeinem der offiziellen Standards entspricht. HTML-Dateien zu schreiben, die wenigstens von den gängigen Browsern auf den gängigen Betriebssystemen identisch angezeigt werden, ist eine Kunst geworden.

HTML 1.0 selbst war noch ein statisches Format, welches auf allen HTML-Browsern in etwa gleich dargestellt wurde. Nunmehr kann man in HTML dynamische Scripts einbauen, die dem selben Dokument je nachdem, auf welchem Rechner oder mit welchem Browser es dargestellt wird, völlig unterschiedlichen Inhalt geben können.

### **E-Mail-Formate (message/rfc822, multipart/...)**

Auch bei der E-Mail ist eine starke Entwicklung in Richtung mehr Funktionalität beobachtbar. Das ursprüngliche Mailformat RFC 822 enthielt im Prinzip nur die Möglichkeit, Texte im Ascii-Format zu versenden: keine Sonderzeichen, keine Formatierung.

Die Entwicklung von MIME ermöglichte es nicht nur, auch Sonderzeichen in E-Mails zu berücksichtigen. Mittels MIME kann das Format der E-Mail auch völlig frei gewählt werden. Der Benutzer kann seine E-Mails auch so versenden, dass er dem Empfänger mehrere Formate zur Auswahl anbietet (z. B. unformatiertes text/plain und formatiertes text/html). Er sollte dann in beide Varianten den selben Inhalt schreiben, aber überprüft kann dies von der Software nicht werden. MIME bietet auch die Möglichkeit, mehrere Dokumente zu einem Paket (multipart) zusammenzufassen (etwa einen Ascii-Text, an den eine Excel-Tabelle angehängt wird).

S/MIME ist das heute am weitesten verbreitete Format zur Versendung signierter oder verschlüsselter E-Mails. Mit S/MIME kann technisch gesehen jeder beliebige Inhalt signiert werden – zum Beispiel ein Paket, das aus einem (alternativ als text/plain oder text/html) angebotenen Text, einer Excel-Tabelle und einer Audiodatei besteht.

### **Formate von Anwendungssoftware (application/...)**

Die größte Zahl existierender Dokumentenformate dient zur Speicherung von Dokumenten, die mit Anwendungsprogrammen erzeugt wurden. Beispielhaft sei das Format von Microsoft Word herausgegriffen (.DOC, application/msword).

Typisch für die Formate von Anwendungsprogrammen ist, dass die Spezifikation des Formates nicht allgemein verfügbar ist. Es ist zwar bei weithin verbreiteter Software wie Microsoft Word den Experten im Prinzip bekannt, wie eine .DOC-Datei aufgebaut ist. Eine offizielle Spezifikation, aus der alle Feinheiten hervorgehen, ist aber nicht öffentlich zugänglich. Dies hängt auch damit zusammen, dass ein Anbieter kommerzieller Software sich von der Konkurrenz nicht in die Karten sehen lassen möchte. Die Software wird ständig

weiterentwickelt und ändert sich nicht nur dann, wenn eine neue Version mit neuer Versionsnummer vermarktet wird. Vielmehr gibt es zahlreiche Zwischenversionen, in denen Fehler behoben und neue kleine Features eingebaut wurden. Dies kann natürlich Änderungen an der Spezifikation des Datenformates bewirken. Unter Umständen nützt der Hersteller auch bloß Möglichkeiten aus, die im Dokumentenformat bereits spezifiziert wurden, die aber – da sie bislang nie verwendet wurden – niemandem bekannt waren.

Ein besonderes Problem hoch entwickelter Dokumentenformate besteht darin, dass diese Formate die Grenze vom Dokumentenformat zur Programmiersprache eigentlich bereits überschritten haben. In einer Word-Datei kann man Visual-Basic-Makros schreiben, die die Funktionalität einer vollwertigen Programmiersprache nutzen. Mit der Makrosprache kann man auch auf beliebige Dateien zugreifen und diese verändern. Mittlerweile existiert daher auch schon eine große Anzahl von Makroviren – also von Makros, die die Eigenschaft haben, sich selbst zu verbreiten.

Es ist nicht besonders schwer, ein Makro zu programmieren, das den Text dynamisch verändert. Noch leichter ist es, dynamische Veränderung mittels Feldfunktionen zu programmieren. Fügt man in einer Word-Datei den folgenden Ausdruck ein: {WENN {AKTUALDAT} = „15.01.00“ 5.000 10.000} (die geschwungenen Klammern erstellt man mit dem Menüpunkt „Einfügen – Feld ...“), dann wird an dieser Stelle am 15.01.2000 der Betrag von 5.000 und an allen Tagen der doppelte Betrag angezeigt. Die Gültigkeit einer Signatur über das Word-Dokument würde dabei nicht beeinträchtigt, da ja die Feldfunktion und nicht der jeweils angezeigte Betrag elektronisch signiert wird.

### **RTF – Rich Text Format (text/rtf)**

RTF ist im Prinzip eine „veröffentlichte“ Variante des Microsoft-Word-Formates. Word-Dateien oder Dateien anderer Textverarbeitungsprogramme können als RTF-Dateien abgespeichert werden.

Auch in RTF lassen sich Word-Makros oder Feldfunktionen abspeichern. Da das Format öffentlich verfügbar ist, ist es aber leichter als beim nicht öffentlich verfügbaren Word-Format selbst, Software zu programmieren, die solche dynamischen Elemente aufspürt und davor warnt oder sie entfernt.

### **Postscript (application/postscript)**

Postscript ist eine weit verbreitete Seitenbeschreibungssprache. Auch für Postscript gilt, dass die Funktionalität so hoch entwickelt ist, dass die Grenze zur Programmiersprache schon überschritten wurde.

Um sicherzustellen, dass ein Postscript-Dokument keine dynamischen oder sicherheitskritischen Elemente enthält, muss man – wie oben bei RTF beschrieben – eine Codeprüfungssoftware auf das Dokument anwenden, die die problematischen Elemente aufspürt.

### **PDF – Portable Document Format (application/pdf)**

PDF ist ein auf Postscript basierendes Dokumentenformat, das zum Ziel hat, auf allen Plattformen identisch dargestellt zu werden. Die Programme zur Anzeige von PDF-Dateien sind kostenlos erhältlich. Der Acrobat Writer zum Erstellen von PDF-Dateien wird von Adobe Systems Inc. produziert und vertrieben.

Die Konvertierung des Dokumentes einer Anwendungssoftware – wie z. B. Microsoft Word – in das PDF-Format hat den Vorteil, dass das Aussehen des Dokumentes dabei „eingefroren“

wird. Dynamische Veränderungen aufgrund von Word-Makros oder Word-Feldfunktionen sind danach nicht mehr möglich. Bei den PDF-Dateien ist auch in höherem Maße gewährleistet, dass das Dokument auf verschiedenen Plattformen gleich aussieht. (Ein Problem von Word besteht z. B. darin, dass der Seitenumbruch oftmals an ganz anderen Stellen erfolgt, wenn man das Dokument auf einem anderen Rechner oder auch nur einem anderen Drucker ausdruckt.) Zu beachten ist aber, dass PDF zwei Arten von Viewern unterstützt. First-Level-Viewer zeigen das Dokument in einer standardisierten Form an. Second-Level-Viewer erlauben den Einsatz von Plug-In-Modulen, um die Funktionalität zu erhöhen. Die Verwendung dieses zweiten Levels durch den Signator kann zur Folge haben, dass der Empfänger der Nachricht diese nicht (vollständig) lesen kann, wenn er nicht über das richtige Plug-In-Modul verfügt.

Ein großer Nachteil der Konvertierung besteht darin, dass sie unumkehrbar ist. Die PDF-Datei kann also nicht in Word weiterbearbeitet werden. Außerdem ist auch bei PDF-Dateien nicht gewährleistet, dass sie auf allen Rechnern gleich aussehen. Der Aufsichtsstelle sind etwa PDF-Dateien bekannt, bei denen die letzten Zeichen jeder Zeile nicht ausgedruckt werden konnten.

Die Aufsichtsstelle hat auch bereits mehrmals festgestellt, dass es bei der Konvertierung von Dokumenten von Word zu PDF zu Informationsverlust kommen kann. In einem Fall verschwanden etwa einzelne Zeilen einer Tabelle im Zuge der Konvertierung hinter Kopf- und Fußzeile des Dokuments. In einem anderen Fall rutschte eine Hintergrundgrafik aus unerklärlichen Gründen in den Vordergrund und verdeckte den Text.

### **XML (text/xml, application/xml)**

Ein neues Format, mit dem auf den zunehmenden Wildwuchs von Dokumentenformaten – insbesondere bei HTML – eingegangen wurde, ist XML. Eigentlich handelt es sich dabei nicht um ein Dokumentenformat, sondern um eine Sprache, in der Dokumentenformate spezifiziert werden können.

Möglicherweise trägt XML dazu bei, dass die Qualität veröffentlichter Spezifikationen verbessert wird. Die Programmierung von Codeprüfungssoftware, die dynamische Elemente oder sonst problematische Stellen aufspüren kann, wäre dadurch erleichtert.

### **Spezialformate für elektronische Signaturen**

Um den Anforderungen an zu signierende Dokumente zu entsprechen, wurden Spezialformate entwickelt, mit denen unerwünschte dynamische Veränderungen oder Unsichtbarkeiten ausgeschlossen werden sollen.

Der Aufsichtsstelle ist etwa das Format Signform bekannt, mit dem Formulare gestaltet werden können. Es sind zwar gewisse dynamische Veränderungen möglich (man kann z. B. ein Formular so gestalten, dass nur dann Informationen zur Bankverbindung abgefragt werden, wenn als Zahlungsform „Bankeinzug“ angekreuzt wurde), die Spezifikation des Formats soll aber sicherstellen, dass immer nur das signiert wird, was auch tatsächlich angezeigt wurde.

Für bestimmte Anwendungen (insbesondere dann, wenn formularmäßig abgefragte Daten automationsunterstützt weiterverarbeitet werden sollen), können solche Spezialformate sicherlich nützlich sein. Es wird aber niemand einen hundertseitigen Vertrag, an dem mehrere Personen gleichzeitig arbeiten, in einer solchen Formularsprache verfassen.

## **Problemstellungen**

### **Spezifikation**

Wie ein gegebenes Dokument darzustellen ist, kann letztlich nur beantwortet werden, wenn man die Spezifikation des Dokumentenformates kennt. Daher verlangt § 7 Abs. 2 des Entwurfs der Signaturverordnung aus gutem Grund, dass nur auf solche Dokumente eine sichere elektronische Signatur angebracht werden darf, deren Spezifikation allgemein verfügbar ist.

Wie oben erwähnt wurde, gilt für die meisten Anwendungsprogramme aber, dass die Spezifikation nicht allgemein verfügbar ist. Das Microsoft-Word-Dokumentenformat gehört zwar zu den verbreitetsten Dokumentenformaten auf der Welt und eine Word-Datei kann von fast jedem Benutzer geöffnet und angezeigt werden – auch von Benutzern anderer Software, die in der Regel über entsprechende Filter verfügt. Die Datei sieht aber nicht überall exakt gleich aus, was zu sinnstörenden Fehlern führen kann. Wie sie auszusehen hat, kann letztlich nur ein Blick in die Spezifikation klären.

Es stellen sich also die Probleme,

- dass eine allgemein verfügbare Spezifikation existieren muss und
- dass die Konvertierung des Dokuments in dieses spezifizierte Dokumentenformat möglich sein muss (ohne dass dabei sinnstörende Veränderungen am Inhalt vorgenommen werden).

### **Dynamische Veränderungen**

Besonders problematisch sind dynamische Veränderungen des Dokumentinhaltes nach Erstellung der Signatur. Es könnte zwar im Streitfall in der Regel leicht festgestellt werden, worauf diese Veränderung zurückzuführen ist (das für die Veränderung verantwortliche Makro bzw. die Feldfunktion können ja nicht ohne Zerstörung der Signatur aus dem Dokument entfernt werden). Die Täuschung oder der Irrtum des Signators sind zu diesem Zeitpunkt aber bereits geschehen und ein Schaden kann dadurch bereits entstanden sein.

Der Entwurf der Signaturverordnung verlangt daher in § 7 Abs. 2, dass – wenn in einem Dokumentenformat dynamische Veränderungen codiert werden können – diese Codierungen nicht verwendet werden dürfen.

In vielen Dokumentenformaten (man denke etwa an die häufig für die Erstellung von Rechnungen verwendeten Tabellenkalkulationsprogramme) wird es aber schwierig sein, softwaretechnisch eine eindeutige Grenze zwischen dynamischen und statischen Elementen zu ziehen. Eine Variante des Dokumentenformates zu spezifizieren, die die Funktionalität aufrecht erhält, aber problematische dynamische Elemente ausschließt, ist daher sicher eine schwierige Aufgabe.

### **Unsichtbarkeiten**

Ähnliche Probleme wie dynamische Veränderungen können durch unsichtbare Daten im signierten Dokument entstehen. Unproblematisch sind dabei sicher jene in allen höher entwickelten Dokumentformaten enthaltenen Formatierungsanweisungen, die dem Benutzer unter gar keinen Umständen angezeigt werden (in der Regel wird der größte Teil des Inhaltes einer Datei von der Software nicht angezeigt, sondern interpretiert), da sich die Rechtswirkungen der Signatur keinesfalls darauf erstrecken können.

Problematisch sind aber jene Fälle, in denen manche Inhalte auf dem einen Rechner angezeigt werden, auf dem anderen aber nicht. Hier kann strittig sein, ob der Text auf dem

Rechner des Signators sichtbar und damit von seiner Willenserklärung umfasst war. Gerade solche Elemente sind von Codeprüfungssoftware aber schwer zu erkennen. Dass weiße Schrift auf hellgrauem Grund ein Problem darstellt, ist ziemlich klar. Weiße Schrift auf färbigem Grund kann auf manchen Rechnern gut lesbar sein, auf anderen (z. B. beim Ausdruck) als weiße Schrift auf weißem Grund dargestellt werden. Und woraus soll die Codeprüfungssoftware erkennen, dass die als `<font size=-3>` gekennzeichnete Schrift auf einem anderen Rechner aufgrund der dort verwendeten Einstellungen des Browsers und des Grafiktreibers so klein dargestellt wird, dass sie nicht einmal mehr als Schrift erkennbar ist?

Es ist daher wahrscheinlich schwierig, ein Dokumentenformat zu spezifizieren, das die Funktionalität eines Textverarbeitungsprogrammes aufrecht erhält, aber dennoch sicherstellt, dass es keine „unsichtbaren“ Elemente gibt.

## Lösungsmöglichkeiten

Der Lösungsansatz des § 7 Abs. 2 des Entwurfs der SigV sieht vor, dass der Zertifizierungsdiensteanbieter dem Signator Formate zu empfehlen hat, die sicher elektronisch signiert werden können. Für die zu signierenden Daten dürfen nur die vom Anbieter empfohlenen Formate verwendet werden. Die Spezifikation dieser Formate muss allgemein verfügbar sein. Können in einem Format auch dynamische Veränderungen oder unsichtbare Daten codiert werden, so dürfen die betreffenden Codierungen nicht verwendet werden. Der Anbieter hat die Anwender anzuweisen oder ihnen Methoden bereitzustellen, um dynamische Veränderungen oder unsichtbare Daten auszuschließen.

## Fragen

- Gibt es beim gegenwärtigen Stand der Technik – abgesehen von simplen Formaten wie Ascii (text/plain) – bereits „sichere“ Dokumentenformate, deren Spezifikation allgemein verfügbar ist und mit denen dynamische Veränderungen und unsichtbare Daten ausgeschlossen oder doch zumindest in ihrer Problematik reduziert werden können – etwa Subvarianten von RTF oder von Postscript?
- Gibt es beim gegenwärtigen Stand der Technik praktikable Konvertierungsprogramme, mittels derer Dokumente aus Standardformaten in solche „sicheren Formate“ konvertiert werden können? Inwieweit besteht die Möglichkeit, Dokumente in „sicheren Formaten“ elektronisch weiterzuverarbeiten?
- Wie sind die Kosten „sicherer“ Dokumentenformate und von Konvertierungs- oder Codeprüfungsprogrammen für die Anbieter, die Signatoren und die Empfänger signierter Nachrichten einzuschätzen?
- Zwischen dem Signator und seinem Zertifizierungsdiensteanbieter gibt es ein vertragliches Naheverhältnis, im Rahmen dessen der Signator vom Anbieter über die Problematik „sicherer“ und „unsicherer“ Dokumentenformate informiert werden kann und in dem der Anbieter den Signatoren Software zur Verfügung stellen kann. Wie aber kann der Empfänger signierter Nachrichten überprüfen, ob es sich um sichere Dokumentenformate handelt und wie erlangt er Zugang zu Software, mit der er diese Überprüfung vornehmen kann?
- Die Signaturverordnung trifft keine Aussage darüber, welche Rechtsfolgen entstehen könnten, wenn der Signator die Pflichten des § 7 Abs. 2 nicht einhält und „unsichere“ Dokumentenformate signiert. Welche Rechtsfolgen könnte dies haben bzw. welche Rechtsfolgen wären im Sinne des Vertrauensschutzes geboten?

## 2. Aufbewahrung der privaten Schlüssel

### Anforderungen

Das Signaturgesetz verlangt für die Erstellung sicherer elektronischer Signaturen den Einsatz solcher technischer Komponenten, die die unbefugte Verwendung von Signaturerstellungsdaten (privaten Schlüsseln) verlässlich verhindern (§ 18 Abs. 1 SigG).

Der wesentlichste Aspekt dieser Anforderung besteht natürlich darin, dass die privaten Schlüssel so aufbewahrt werden, dass sie von einem Angreifer nicht ausgelesen werden können. Da dies mit herkömmlichen Betriebssystemen unmöglich gewährleistet werden kann, ist die derzeit praktikabelste Lösung, die privaten Schlüssel auf einer Chipkarte zu speichern. Das spezielle Betriebssystem einer solchen Chipkarte stellt dann sicher, dass die privaten Schlüssel die Chipkarte niemals verlassen. Der Chipspeicher soll auch mit anderen Methoden nicht gebrochen werden können – man soll also weder unter dem Mikroskop, noch durch ausgefeilte Analyse des Stromverbrauchs der Karte (DPA) oder von Fehlfunktionen der Karte unter Extrembedingungen (DFA) irgendwelche Rückschlüsse auf den Schlüssel ziehen können.

Die im Entwurf vorliegende Signaturverordnung führt diese Anforderungen an mehreren Stellen detaillierter aus:

- Erzeugt der Anbieter die Schlüssel selbst, so muss er geeignete Vorkehrungen treffen, die die Speicherung des privaten Schlüssels außerhalb der Chipkarte ausschließen (§ 3 Abs. 6). Das Erzeugungssystem und die Übertragung der Schlüssel auf die Chipkarte sind entsprechend zu überwachen.
- Werden die Schlüssel in der Chipkarte erzeugt, so muss die Chipkarte technisch geeignet sein (§ 3 Abs. 7).
- Das Duplizieren privater Schlüssel nach der Erzeugung ist untersagt (§ 4 Abs. 1).
- § 9 verlangt für die Prüfung der technischen Komponenten die Einhaltung bestimmter technischer Normen – in diesem Zusammenhang sind dabei insbesondere ITSEC E3 „hoch“ und FIPS 140-1, level 2, von Relevanz.

In engem Zusammenhang damit steht eine Reihe von technischen Anforderungen, die sich nicht auf die Aufbewahrung der privaten Schlüssel, aber auf ihre mathematischen Eigenschaften beziehen:

- Nicht jedes Verfahren, sondern nur bestimmte kryptographische Verfahren sind für die Erstellung sicherer elektronischer Signaturen zugelassen (§ 3, Anhänge 1 und 2).
- Die Schlüssel müssen eine bestimmte Mindestlänge haben, damit man aus dem öffentlichen Schlüssel nicht den privaten Schlüssel errechnen kann (§ 3 Abs. 3).
- Die Schlüssel müssen auf einer tatsächlichen Zufälligkeit beruhen (§ 3 Abs. 5).

Die wesentlichste Anforderung der Signaturverordnung besteht in diesem Zusammenhang darin, dass die geforderten Eigenschaften nicht nur erfüllt sein müssen, sondern dass dies auch nach standardisierten Kriterien von einer unabhängigen Stelle überprüft werden muss. Die maßgeblichen Normen zur Evaluation sind dabei FIPS 140-1 und ITSEC.

ITSEC ist eine allgemein gehaltene Norm, nach der praktisch alle Sicherheitsfragen in der Informationstechnik evaluiert werden können. Welche Sicherheitsvorgaben tatsächlich geprüft werden, wird daher nicht vom ITSEC-Standard selbst vorgegeben, sondern von demjenigen, der die Prüfung in Auftrag gibt. Zusätzlich zur Erfüllung der Evaluationsstufe E3 mit der Mindeststärke der Sicherheitsmechanismen „hoch“ wird man also im Hinblick auf die ITSEC-Evaluierung verlangen müssen, dass die oben genannten Anforderungen der Signaturverordnung als Sicherheitsvorgaben für die Evaluierung maßgeblich waren.

Der Umstand, dass die Ausstellung qualifizierter Zertifikate an eine entsprechend sichere Aufbewahrung des privaten Schlüssels gebunden ist, bedeutet für den Empfänger einer signierten Nachricht einen großen Sicherheitsvorteil. Er kann sich zumindest sicher sein (und kann dies aus der signierten Nachricht bzw. dem Zertifikat des Signators erkennen), dass für die Erstellung der Signatur eine Chipkarte verwendet wurde, die der Signator, als seine Identität vom Anbieter geprüft wurde, unter seiner alleinigen Kontrolle hatte. Dies ist nach dem Stand der Technik geprüft und sichergestellt. Der Signator kann nicht behaupten, dass sich jemand durch einen geschickten Angriff unbemerkt Zugriff auf den privaten Schlüssel verschafft hat. Ein Dritter kann den privaten Schlüssel nur verwenden, wenn er Zugriff auf die Chipkarte hat und ihren PIN-Code kennt.

## Fragen

- Soweit der Aufsichtsstelle bekannt ist, gibt es am Markt bereits eine Reihe von Chipkarten, die nach ITSEC bzw. nach FIPS 140-1 evaluiert wurden. Decken diese Evaluierungen alle Anforderungen der Signaturverordnung ab?
- Sind in diesem Zusammenhang Sicherheitsprobleme bekannt, die noch nicht in ausreichender Weise gelöst wurden?
- Inwieweit gibt es bereits Alternativlösungen zu Chipkarten, die geeignet wären, die Anforderungen des Signaturgesetzes und der Signaturverordnung zu erfüllen?
- Wie sind die Kosten der jeweiligen Möglichkeiten, private Schlüssel sicher zu verwahren, einzuschätzen?

## 3. Kontrolle des Signiervorganges

### Anforderungen und Gefahren

Durch die Speicherung der privaten Schlüssel auf einer Chipkarte oder einem ähnlich sicheren Speichermedium ist gegenüber reinen Softwarelösungen ein bedeutsamer Sicherheitsfortschritt erreicht. Jedenfalls ist sichergestellt, dass die Signatur ohne die Chipkarte nicht ausgelöst werden kann.

Schwieriger ist es aber, dem Signator tatsächlich vollständige Kontrolle über den Signaturvorgang zu gewähren und „die unbefugte Verwendung von Signaturerstellungsdaten verlässlich [zu] verhindern“ (§ 18 Abs. 1 SigG).

Die zu signierenden Daten werden natürlich nicht auf der Chipkarte erstellt und auch nicht von der Chipkarte dem Benutzer angezeigt. Im Regelfall wird man den zu signierenden Text in einem Textverarbeitungsprogramm oder E-Mail-Programm erstellen und ihn dann zur Signatur an eine Signatursoftware weiterleiten, die ihrerseits den Hashwert bildet und diesen an die Chipkarte zur Signatur weiterleitet. Auf dem Weg bis zur Chipkarte könnten die Daten oder der Hashwert verändert werden.

Ein weiteres Problem besteht darin, dass die Chipkarte durch einen entsprechenden Authentifizierungsmechanismus entriegelt werden muss. Dies kann z. B. ein PIN-Code sein oder aber ein Fingerabdruck, den eine mit biometrischer Funktionalität ausgestattete Karte erkennt. Kann eine böse eingeschleuste Software den PIN-Code abfangen, dann kann sie in weiterer Folge beliebig viele zu signierende Daten an die Chipkarte senden. Die Chipkarte kann ja nicht erkennen, ob der PIN-Code vom legitimen Benutzer eingegeben wurde oder ob er von einem böseartigen Programm gesandt wird.

Ein möglicher gezielter Angriff könnte z. B. darin bestehen, dass ein Hacker seinem Opfer ein mit dem Hacker-Tool BackOrifice infiziertes Programm zusendet. BackOrifice ist ein Programm, das sich in Windows-Betriebssysteme einklinkt und dabei eine praktisch vollständige Überwachung des Rechners von außen ermöglicht. Das Programm lässt sich leicht an andere, harmlos aussehende Programme (etwa die manchmal als Geburtstagsglückwünsche versandten E-Mails, die ein Programm mit einer lustigen animierten Grafik enthalten) anhängen – arbeitet also als sogenannter „Trojaner“. War das Opfer so unvorsichtig, das Programm zu starten, so kann der Hacker unbemerkt alle Tastatureingaben überwachen, bis er den PIN-Code weiß. Dann kann er die Signatursoftware starten und mit Hilfe des PIN-Codes signieren, solange sich die Chipkarte im Chipkartenleser befindet. (Ein Auslesen der Chipkarte ist aber – wenn die in Kapitel 2 genannten Anforderungen erfüllt sind – nicht möglich. Der Hacker ist also machtlos, sobald die Chipkarte aus dem infizierten Rechner entfernt wird.)

### Lösungsmöglichkeiten

Solche oder andere Angriffe hundertprozentig zu verhindern, ist praktisch unmöglich. Man kann ein einzelnes System sehr gut schützen, wenn man sich entsprechend ausführlich mit den Sicherheitsanforderungen und Gefahren auseinandersetzt. Dass ein Zertifizierungsdiensteanbieter aber ein Sicherheitskonzept für die Signatoren erstellt, das sowohl maximale Sicherheit gewährleistet als auch die nötige Flexibilität bietet, um unterschiedlichen Bedürfnissen einer großen Zahl von Signatoren zu entsprechen, ist nicht möglich. Jeder Schritt, der die Sicherheit erhöht, vermindert die Funktionalität.

Die Konsultation soll aber dazu beitragen, Möglichkeiten aufzuzeigen, die der Anbieter den Signatoren empfehlen oder bereitstellen kann, um die Sicherheit zu erhöhen. Einige solcher Möglichkeiten wären etwa die folgenden:

**Auswahl des Betriebssystems:** Ein Grund dafür, dass die oben geschilderte Attacke erfolgreich sein kann, besteht darin, dass der Benutzer ein Betriebssystem verwendet, das für die Benutzung durch jeweils einen einzelnen Benutzer entwickelt wurde und daher nur über eine eingeschränkte Berechtigungsverwaltung verfügt. Unter Windows 95 etwa kann jeder Benutzer auf die Dateien des Betriebssystems zugreifen. Startet er unvorsichtiger Weise ein mit einem Trojaner infiziertes Programm, so kann sich der Trojaner in den Tiefen des Betriebssystems einnisten. Betriebssysteme, die über eine stark ausgeprägte Berechtigungsverwaltung verfügen – wie Unix oder Windows NT – erlauben eine Konfiguration, bei der nur ein Administrator, nicht aber ein gewöhnlicher Benutzer berechtigt ist, Änderungen im Betriebssystem vorzunehmen. Eine solche Konfiguration zu installieren, ist aber nicht trivial und kann von den meisten Benutzern nicht selbst durchgeführt werden.

**Auswahl der Anwendersoftware:** Eine Empfehlung, die der Anbieter dem Signator jedenfalls geben kann (und soll), besteht darin, dass der Signator nur vertrauenswürdige Software auf dem Rechner installieren soll. Problematisch ist aber, dass gerade auch die Software namhafter Hersteller gravierende Sicherheitsprobleme aufweisen kann. Das ständige Bemühen, durch die Entwicklung neuer Features Marktanteile zu erringen, kann dazu führen, dass auch in Standardsoftware schwere Sicherheitsmängel Eingang finden. Als Beispiel könnte etwa der Virus Bubble Boy genannt werden, der in den Betriebssystemen Windows 98 und Windows 2000 Sicherheitslücken im Programm Microsoft Outlook (Express) gemeinsam mit dem Internet Explorer 5 ausnützen konnte. Dem – als E-Mail versandten – Virus war es unter bestimmten Umständen möglich, in das Betriebssystem einzudringen, ohne dass der Empfänger die E-Mail auch nur aktiv geöffnet oder gelesen hätte.

**Verwendung von Virenprüfprogrammen:** Vielfach kann auf den Rechner eingeschleuste Software von Virenprüfprogrammen erkannt werden – nicht nur dann, wenn es sich um einen Virus im eigentlichen Sinn des Wortes (nämlich um ein Programm, das sich selbst vermehren kann) handelt. Virenprüfprogramme können aber nur gegen solche Software schützen, die den einschlägigen Forschungszentren bekannt ist – nicht aber gegen spezialisierte Attacken auf einen bestimmten Rechner.

**Signatursoftware:** Die Signatursoftware könnte über spezielle Sicherheitsfunktionen verfügen. Beispielsweise könnte die Kommunikation zwischen der Chipkarte und der Signatursoftware so gestaltet sein, dass die Chipkarte die Signatursoftware erkennt und es nicht zulässt, wenn sie von anderen Programmen angesteuert wird. – Die Formulierung in § 9 Abs. 2 des Entwurfs der SigV legt es nahe, dass die Signatursoftware auf die Einhaltung solcher Sicherheitsvorgaben geprüft werden sollte. Welche Sicherheitsvorgaben einer solchen Evaluierung zu Grunde gelegt werden sollen, bleibt aber unklar. Außerdem könnte eine solche Evaluierung auch keine Sicherheit gegen Attacken wie die oben beschriebene bieten. Wenn es ein Hacker schafft, sich mittels BackOrifice Zugriff auf den Computer zu verschaffen, dann braucht er die Signatursoftware nicht zu umgehen, sondern kann sie einfach starten.

**Eingabe des PIN-Codes in eine Spezialtastatur/Biometrie:** Chipkartenleser werden wahrscheinlich in Zukunft in die Tastatur eingebaut werden. Dabei besteht die Möglichkeit, die Tastatur so zu gestalten, dass der PIN-Code direkt über die Tastatur an die Chipkarte gesendet wird, ohne dass er an das Betriebssystem des Rechners gelangt. Er bleibt damit auch für einen Hacker, der sich Zugriff auf das Betriebssystem verschaffen konnte, unsichtbar. Noch sicherer sind biometrische Karten. Bei diesen scheiden nicht nur Hackerattacken auf den PIN-Code aus. Sie sind auch dagegen geschützt, dass der Signator

selbst sie an einen Dritten zur Signaturerstellung weitergibt. Den PIN-Code könnte er weitersagen, wenn er selbst nicht signieren will. Einen Fingerabdruck kann er aber nicht delegieren.

## Fragen

- Inwiefern ist es beim gegebenen Stand der Technik möglich, die unbefugte Verwendung von privaten Schlüsseln verlässlich zu verhindern? Für wie praktikabel werden die oben dargestellten Sicherheitsmaßnahmen eingeschätzt bzw. welche anderen Sicherheitsmaßnahmen gibt es?
- Es ist klar, dass jeder einzelne Signator die Sicherheit seines eigenen Rechners nach seinen Bedürfnissen nach Belieben steigern kann – je nachdem wie viel Aufwand er dafür betreibt und zu welchem Verlust an Funktionalität er bereit ist. Inwieweit gibt es aber praktikable Sicherheitsmaßnahmen, die sich standardisieren und auf eine größere Zahl von Signatoren anwenden lassen?
- Aus der Sicht des Empfängers einer signierten Nachricht betrachtet, ist vor allem interessant, ob Sicherheitsmängel dem Signator zugerechnet werden können (gibt dieser z. B. die Chipkarte samt PIN-Code weiter, was sich vom Zertifizierungsdiensteanbieter nicht verhindern lässt, dann haftet der Signator dennoch). Inwieweit können Sicherheitsmaßnahmen es sicherstellen, dass die Kompromittierung der Sicherheitsmaßnahmen dem Signator zugerechnet werden kann?
- Eine weitere Steigerung der Sicherheit aus der Sicht des Empfängers einer signierten Nachricht besteht darin, dass gewisse Sicherheitsmaßnahmen vom Signator nicht einmal dann außer Kraft gesetzt werden können, wenn er es wünscht (beispielsweise kann der Signator selbst an den eigenen privaten Schlüssel nicht heran, wenn dieser in einer entsprechenden Chipkarte gespeichert ist). Gibt es in diesem Zusammenhang weitere mögliche Sicherheitsmaßnahmen?
- Es wird diskutiert, dass die sichere elektronische Signatur auch Personen zur Verfügung stehen soll, die nicht über einen eigenen PC und Internetzugang verfügen. Diese Personen könnten an öffentlichen Terminals – etwa in Gemeindeämtern oder Banken – am elektronischen Rechts- und Geschäftsverkehr teilnehmen. Welche Anforderungen müsste man an solche Terminals stellen bzw. wie könnte der Signator erkennen, ob er dem Gerät vertrauen kann?
- Wie sind die Kosten der jeweils vorgeschlagenen Sicherheitsmaßnahmen einzuschätzen?

## 4. Verwendung von Schlüsseln für andere Zwecke

Die für die Signatur eingesetzten kryptographischen Verfahren eignen sich nicht nur zur Signatur, sondern auch für andere Zwecke wie Verschlüsselung, Authentifizierung oder zur Sicherung von Finanztransaktionen. Aus den unterschiedlichen Einsatzzwecken ergeben sich unterschiedliche Anforderungen an die Chipkarte und die Signatursoftware. Manche Einsatzzwecke müssen auch dadurch unterschieden werden, dass unterschiedliche Schlüsselpaare dafür verwendet werden.

### Anforderungen

**Rechtswirkungen:** Obwohl es sich bei der Anwendung eines privaten Schlüssels immer um denselben technischen Vorgang handelt, sind mit dem Vorgang verschiedene Rechtswirkungen verbunden. Mit der elektronischen Signatur gibt der Anwender in der Regel eine nach außen hin wirksame Willenserklärung ab. Bei der Authentifizierung will er nur seine Identität bekunden. Beim Entschlüsseln will er überhaupt nicht nach außen hin auftreten. Welchen Vorgang er gerade auslöst, muss für den Anwender erkennbar sein, da sonst die mit der Signatur verbundenen Rechtswirkungen nicht begründet werden könnten. § 7 Abs. 3 des Entwurfs der SigV verlangt daher, dass derselbe PIN-Code nicht für unterschiedliche Anwendungen (wie z. B. Signatur- und Bankomatfunktion) verwendet werden darf.

**Automatisierung:** Je nach Anwendungszweck wünscht der Anwender unterschiedliche Grade der Automatisierung. Für die Signaturerstellung verlangt § 7 Abs. 3 des Entwurfs der SigV, dass die Anzahl der Signaturen, die mit einer Autorisierung des Signators ausgelöst werden, dem Signator bekannt gegeben wird. Eingabeerleichterungen bei wiederholter Eingabe von Autorisierungs-codes müssen ausgeschlossen sein. Für die Entschlüsselung oder Authentifizierung könnte der Benutzer aber eine Automatisierung wünschen, damit er nicht bei jedem Zugriff auf ein Dokument seinen PIN-Code eingeben muss. Manche Anwendungen sind mit der Signaturerstellung verwandt, dennoch ist aber eine Automatisierung wünschenswert. Dies gilt etwa für Zeitstempeldienste, die ohne Automatisierung kaum Sinn ergeben, oder für die automatisierte Ausstellung signierter Empfangsbestätigungen.

**Backup:** Bei manchen Anwendungen ist ein Backup der privaten Schlüssel wünschenswert, bei anderen Anwendungen ist dies nicht notwendig oder sogar ausgeschlossen. Für die sichere elektronische Signatur verbietet § 4 des Entwurfs der SigV die Duplizierung privater Schlüssel. Verwendet man asymmetrische Kryptographie für die dauerhafte Verschlüsselung von Daten (nicht bloß für die Transportverschlüsselung), dann ist ein Backup des privaten Schlüssels unbedingt erforderlich, um Datenverlust zu vermeiden.

**Anwendung privater Schlüssel auf Zufallszahlen:** Ein Problem besteht darin, dass manche Anwendungen vom Benutzer die Signatur von zufälligen Daten verlangen. Wird der Benutzer etwa bei Verwendung einer SSL-Verbindung im Internet aufgefordert, sich zu identifizieren, so geschieht dies, indem ihm eine Zufallszahl zur Signatur vorgelegt wird. Der Server erkennt dann aus der signierten Zahl, dass der Benutzer tatsächlich Kontrolle über den privaten Schlüssel hat – dass es sich also um die Person handelt, deren Identität im Zertifikat bescheinigt wurde. Ein privater Schlüssel, mit dem rechtlich relevante Willenserklärungen signiert werden sollen, darf für SSL-Authentifizierung aus dem folgenden Grund nicht verwendet werden. Der Server könnte dem Benutzer statt einer Zufallszahl den Hashwert eines Vertrages unterschieben, und der Benutzer würde, indem er sich identifiziert, den Vertrag rechtsgültig signieren. – Ein ähnliches Problem stellt sich, wenn man asymmetrische Schlüsselpaare bei der Verschlüsselung nicht nur zum Key-Agreement verwendet, sondern die gesamte Nachricht asymmetrisch verschlüsselt (z. B. mit RSA in

ECB-Mode). Man könnte einem Benutzer dann eine Signatur unterschreiben, indem man einen Hashwert in ein Dokument einbettet, das man dem Benutzer verschlüsselt übersendet. Mit dem Entschlüsseln (= Anwendung des privaten Schlüssels auf die Daten) würde der Empfänger den Hashwert signieren. Dieses konstruierte Beispiel dürfte aber in der Praxis wenig relevant sein, da asymmetrische Verfahren in der Regel ausschließlich für das Key-Agreement, nicht aber für die Verschlüsselung der Nachricht selbst eingesetzt werden.

Unterschiedliches Sicherheitsniveau: Aus der Sicht des Benutzers besteht ein Unterschied, ob er die Chipkarte auf seinem eigenem PC benutzt oder etwa auf einem öffentlichem Terminal oder in einer Arztpraxis. (Es gibt z. B. Diskussionen, eine Sozialversicherungs-Chipkarte auszugeben, mit der der Patient in der Arztpraxis dem Arzt Zugriff auf die Krankengeschichte einräumt oder andere Berechtigungen erteilt.) Der Benutzer wird den PIN-Code, den er für sichere elektronische Signatur nutzt, unter Umständen nicht in ein Gerät eintippen wollen, das aus seiner persönlichen Sicht nicht sicher genug ist.

## Lösung

Die Lösung der dargestellten Anforderungen besteht darin, dass der Benutzer jeweils für unterschiedliche Anwendungen unterschiedliche Schlüssel und unterschiedliche PIN-Codes verwendet. Die Anzahl der vom Benutzer zu verwaltenden Schlüssel soll aber nicht zu groß sein, damit er nicht den Überblick darüber verliert. Verlangt man vom Benutzer die Verwaltung von fünf Schlüsseln mit unterschiedlichen PIN-Codes, so wird er die PIN-Codes alle auf die Chipkarte schreiben, um sie nicht zu vergessen.

- Wie viele Schlüsselpaare sollte man typischerweise verwalten, um die wesentlichen Funktionen abzudecken (z. B. eines für sichere elektronische Signatur – also starke Rechtswirkungen, eines für Verschlüsselung und ein drittes für Authentifizierung und automatisierte Signaturen mit schwachen Rechtswirkungen, z. B. für automatisierte Empfangsbestätigungen)?
- Wie kann der Anbieter verhindern, dass der Signator seine Schlüssel zweckwidrig nutzt? Der X.509-Standard erlaubt die Angabe der keyUsage im Zertifikat. Gibt es darüber hinaus praktikable Methoden?

## 5. Sichere Signaturprüfung

§ 18 Abs. 4 SigG stellt eine Reihe von Anforderungen an die technischen Komponenten und Verfahren, mit denen eine sichere elektronische Signatur überprüft werden kann. Im Wesentlichen handelt es sich bei diesen Anforderungen um das Spiegelbild der Anforderungen an die Signaturerstellung. Die Anforderung, dass die signierten Daten nicht verändert worden sind, ist etwa durch die Anforderungen an die bei der Signaturerstellung eingesetzten kryptographische Verfahren und an das Dokumentenformat erfüllt. Die für die Signaturerstellung eingesetzten Komponenten werden daher in der Regel auch für Signaturprüfung geeignet sein.

Ein wesentlicher Unterschied ergibt sich aber daraus, dass Signaturen nicht nur von solchen Personen überprüft werden sollen, die selbst als Signatoren auftreten. Eine elektronische Signatur soll vielmehr von einem wesentlich größeren Personenkreis überprüfbar sein. Im Prinzip ist es auch nicht notwendig, dass der Empfänger einer elektronischen Signatur über eine spezielle technische Ausstattung verfügt. S/MIME-signierte E-Mails können etwa mit jedem gängigen E-Mail-Programm überprüft werden.

- Aus § 9 Abs. 2 und § 7 Abs. 5 des Entwurfs der SigV ergibt sich ein Unterschied zwischen einer „normalen“ Signaturprüfung und einer „sicheren“ Signaturprüfung, welche nach ITSEC E3 „hoch“ evaluiert wurde. Inwieweit besteht ein Bedarf an einer solchen „sicheren“ Signaturprüfung und gibt es technische Komponenten, die eine sichere Signaturprüfung vornehmen und ITSEC E3 „hoch“ evaluiert sind?
- Noch sehr schlecht scheint bei den meisten Programmen die Überprüfung der Gültigkeit der Zertifikate zu funktionieren. Das automatisierte Nachschlagen in Verzeichnissen und Überprüfen von CRLs sowie das automatisierte Überprüfen von Zertifikathierarchien ist noch kaum implementiert. Die Überprüfung der Gültigkeit des Zertifikates ist daher meist händisch vorzunehmen. Inwieweit gibt es bereits Produkte, bei denen die Überprüfung automatisiert abläuft?

## 6. Weitere Fragen

In welchem Maße wird ein Bedarf an der Lösung anderer Sicherheitsfragen gesehen, der durch vorhandene Produkte nicht gedeckt ist? Die Antworten auf diese Fragen dienen in erster Linie nicht der Tätigkeit der Aufsichtsstelle, sondern sollen das Ergebnis der Konsultation abrunden, um ein vollständiges Bild über einschlägige Sicherheitsfragen auf den Geräten der Benutzer zu schaffen.

- Wie stellt sich der Bedarf nach Verschlüsselungssoftware dar (Transportverschlüsselung, verschlüsselte Aufbewahrung von Dokumenten)? Welche Anforderungen sind an Produkte zur Verschlüsselung zu richten?
- Um signierte Dokumente in großer Zahl sicher aufbewahren zu können, werden spezielle Dokumentenverwaltungssysteme notwendig sein. Das Dokumentenformat soll auch noch nach langer Zeit lesbar sein. Die Dokumente sollen vor unabsichtlicher Zerstörung der Signatur geschützt sein (die Signatur kann z. B. zerstört werden, wenn ein signiertes Dokument geöffnet, ausgedruckt und wieder gespeichert wird, und die Anwendungssoftware dabei ein neues Datum „zuletzt gedruckt am:“ einfügt). Wie stellt sich der Bedarf nach Dokumentenverwaltungssystemen dar und welche Anforderungen werden an solche Produkte zu richten sein?