Benutzerüberwachte Erzeugung von DSA-Schlüsseln in Chipkarten

Bodo Möller
Universität Hamburg
Fachbereich Informatik
<bmoeller@acm.org>

Zusammenfassung

Bei Chipkarten zum digitalen Signieren soll oft auch der legitime Karteninhaber keine Möglichkeit haben, den geheimen Signierschlüssel zu erfahren. Dieser Beitrag stellt ein effizientes Verfahren zur DSA-Schlüsselerzeugung vor, das sich an Simmons' Protokoll zum Vermeiden von verdeckten Kanälen bei DSA anlehnt: Der Signierschlüssel wird durch ein Zusammenwirken von Kartenausgeber und Karte einerseits und dem Karteninhaber andererseits so erzeugt, daß schon das korrekte Verhalten *einer* Seite ausreicht, um die Qualität des resultierenden Schlüssels sicherzustellen. Diese Methode bringt die Schlüsselerzeugung in den Einflußbereich des Karteninhabers, ohne die Kapselung des Schlüssels in der Chipkarte aufzugeben.

1 Einführung

Die Konzeption von Chipkarten (Smart-Cards; siehe zum Beispiel [GUQ 1992]), die zum digitalen Signieren ([MvOV 1997], [DSA]) von Daten gedacht sind (»Signierkarten«), sieht oft vor, daß es keine Möglichkeit geben soll, den geheimen Signierschlüssel auszulesen – auch nicht für den legitimen Inhaber der Karte, der damit signiert. Ein Grund für diese Kapselung des Geheimschlüssels in der Karte ist die Schadensbegrenzung im Falle z. B. von »trojanischen Pferden« in der beim Signieren benutzten Systemumgebung: »Bösartige« Software kann zwar möglicherweise bei der Karte Signaturen für andere Daten einholen, als der Benutzer glaubt, oder kann ohne dessen Wissen zusätzliche Signaturen erzeugen lassen; bei einem Auslesen des Signierschlüssels aber wäre es darüberhinaus *auch in Zukunft* möglich, *beliebig viele* weitere digitale Signaturen im Namen des Karteninhabers zu berechnen. Als weiterer Grund wird oft ins Feld geführt, daß ein unehrlicher Karteninhaber sonst u. U. seinen eigenen Geheimschlüssel veröffentlichen würde, um dann mit der Behauptung,

der Schlüssel sei offenbar von jemand anderem »geknackt« worden, die Gültigkeit eigener digitaler Signaturen bestreiten zu können.

Unter diesen Voraussetzungen kann auch die Schlüsselerzeugung offensichtlich nicht selbständig vom Karteninhaber durchgeführt werden, da dies der Kapselung des Schlüssels in der Karte entgegenlaufen würde.

Statt dessen könnte die Erzeugung des geheimen Schlüssels z. B. zentral bei einer Kartenausgabestelle erfolgen, und diese könnte dabei auch gleich ein Schlüsselzertifikat [MvOV 1997] produzieren, welches die Zugehörigkeit des (öffentlichen Teils des) Signierschlüssels zum Karteninhaber bestätigt. Dieses Vorgehen ist jedoch problematisch: Die Signierkarte mit ihrem Geheimschlüssel ist organisatorisch nicht dem Karteninhaber zuzuordnen; dieser hat nämlich keine Kontrolle über die Entwurfsvorgänge, kann deren Resultate nicht nachprüfen[1] und kann nicht überwachen, ob die Schlüsselerzeugung tatsächlich sachgemäß durchgeführt wird und der Geheimschlüssel dabei effektiv vor »Ausspähen« durch andere sowie vor Speicherung beim Kartenausgeber geschützt ist. Vielmehr wird die Karte ausschließlich vom Kartenhersteller und Kartenausgeber beeinflußt; die Kontrollmöglichkeit des Karteninhabers beschränkt sich darauf, daß er nach Benutzung der Karte nachprüfen kann, ob die von der Karte erzeugten Signaturen tatsächlich auf die jeweils von ihm vorgegebenen zu signierenden Dateien passen. Über die Funktionsweise der Signierkarte erfährt er dabei so gut wie nichts, eventuelle Hintertüren (z. B. [YY 1996], [YY 1997]) oder Implementierungsmängel (dazu siehe z. B. [BGM 1997]) könnte er nicht bemerken. (Aus diesem Grund brächte es auch keine wesentliche Verbesserung, den geheimen Signierschlüssel direkt auf der Karte mit Hilfe eines eingebauten Zufallsgenerators zu erzeugen statt bei der Kartenausgabestelle.)

Die Karte mit dem Schlüssel hat aus Sicht des Benutzers also *»Black-Box«*-Charakter; sie soll aber trotzdem *»in Vollmacht«* des Karteninhabers Signaturen erzeugen. Der Signierschlüssel ist als ein Schlüssel *der Karte* (und damit des Karten*ausgebers),* nicht als ein persönlicher Schlüssel des Karten*inhabers* anzusehen.

Wenn ein Karteninhaber leugnet, eine laut seinem öffentlichen Schlüssel gültige Signatur selbst mit seiner Karte erzeugt zu haben, dann kann es erstens sein, daß er ganz einfach lügt (oder sich irrt); zweitens ist es aber auch möglich, daß die Signatur in der Tat ohne sein Zutun erfolgt ist – zum Beispiel,

- weil jemand zufällig den geheimen Schlüssel oder auch nur eine gültige Signatur erraten hat (was aber bei sinnvollen Signaturverfahren quasi unmöglich ist);

[1] Zum Sicherheitskonzept der Smart-Cards gehört in der Regel auch, daß deren genaue Funktionsweise für potentielle Angreifer nicht nachvollziehbar sein soll. [AK 1996]

- weil das Signaturverfahren entgegen den Erwartungen unsicher ist[2];
- weil es jemandem, der physischen Zugriff auf die Chipkarte hat, trotz aller Sicherheitsvorkehrungen gelungen ist, den Schlüssel auszulesen [AK 1996];
- oder weil schon Mängel – seien es versehentliche Unzulänglichkeiten oder absichtlich geschaffene »Hintertüren« – beim Kartenentwurf und/oder bei der Schlüsselerzeugung die Sicherheit des Benutzerschlüssels kompromittiert haben.

In diesem Beitrag wird ein Verfahren für DSS-konforme Signaturen [DSA] vorgestellt, das die letztgenannte mögliche Ursache für das »Abhandenkommen« des Geheimschlüssels beseitigt: An die Stelle der zentralen Schlüsselerzeugung tritt ein *Zusammenwirken* von Kartenausgeber und Karte einerseits und dem Karteninhaber andererseits, bei dem schon das korrekte Verhalten *einer* Seite ausreicht, um die Qualität des resultierenden Schlüssels sicherzustellen. Der endgültige Geheimschlüssel wird innerhalb der Karte berechnet; solange die Sicherheitsmaßnahmen der Karte nicht überwunden werden, kann er weder vom Karteninhaber noch vom Kartenausgeber in Erfahrung gebracht werden: Bei korrektem Verhalten der einen Seite läuft eventuelles Fehlverhalten der anderen nur darauf hinaus, das Vorankommen völlig zu blockieren. Dabei wird angenommen, daß die Karte zu einem bestimmten Zeitpunkt dem späteren Benutzer ausgehändigt wird und danach sämtliche Kommunikation zwischen Kartenausgeber und Karte von diesem Karteninhaber überwacht werden kann. (Falls die Signierkarte abhandenkommt, ist diese Voraussetzung nicht mehr gegeben. In dem Fall beruht die Geheimhaltung des Signierschlüssels notwendig auf den Sicherheitseinrichtungen der Chipkarte, in der er gespeichert ist.)

Bei Disputen über laut Überprüfungsalgorithmus gültige Signaturen stellt sich allgemein die Frage, inwieweit dem Karteninhaber tatsächlich die zu »seinem» Schlüssel passenden Signaturen zugerechnet werden können. Wenn der Karteninhaber bestreitet, eine bestimmte Signatur erzeugt zu haben, kann ein Prozeßgegner argumentieren, nach dem Prinzip des Anscheinsbeweises (Beweis des ersten Anscheins, *prima-facie*-Beweis) [Petri 1997] könne auch ohne echten Beweis als typischer Geschehensablauf unterstellt werden, daß der Karteninhaber entgegen seinen Beteuerungen für die Signatur verantwortlich sei. An Anscheinsbeweise stellt der deutsche Bundesgerichtshof zu Recht hohe Anforderungen:

> »Selbst ein nur noch sehr geringes Restrisiko genügt [...] zur Begründung eines Anscheinsbeweises dann nicht, wenn

[2] Dieses Problem läßt sich mit *Fail-Stop*-Signaturverfahren [PP 1997] angehen.

> der Beweisführer selbst durch geeignete und zumutbare Maßnahmen im Vorfeld seine Beweissituation verbessern kann.« [Rüßmann 1998], nach BGHZ 24, 308 ff.
>
> »Wer die Gegenpartei schuldhaft in der Möglichkeit beschneidet, den Anscheinsbeweis zu erschüttern oder zu widerlegen, kann sich nicht auf die Grundsätze des Anscheinsbeweises berufen.« [BGH 1998]

Auf den Fall der Signierkarten angewendet bedeutet das: Da eine zentrale Schlüsselerzeugung ohne Einfluß- und Überwachungsmöglichkeit für den Karteninhaber diesen einem potentiell erhöhten Risiko aussetzt, Opfer von Fehlern oder Betrug zu werden, steht sie bei strenger Beachtung der Grundsätze aus den BGH-Entscheidungen der gerichtlichen Verwertbarkeit der Signaturen entgegen.

In Abschnitt 2 wird der Digital Signature Algorithm (DSA) in seiner Grundform vorgestellt.

Abschnitt 3 stellt ein Verfahren von Simmons vor, das bei DSA-Signaturen mögliche »verdeckte Kanäle« beseitigen kann.

Abschnitt 4 zeigt, wie der Karteninhaber überdies bei der Erzeugung seines Signierschlüssels in der Karte mitwirken und sie überwachen kann.

Abschnitt 5 setzt das in diesem Beitrag vorgestellte Verfahren in Beziehung mit einem Vorschlag aus [FJPP 1995], dem ein vergleichbares Entwurfsziel zugrundeliegt.

2 Der Digital Signature Algorithm (DSA)

Bei dem im Digital Signature Standard [DSA] definierten Digital Signature Algorithm werden folgende Parameter öffentlich bekanntgemacht und können für die Schlüssel mehrerer Teilnehmer verwendet werden:

- Eine Primzahl p mit einer Länge von L Bits (d. h. mit $2^{L-1} < p < 2^L$), wobei L ein Vielfaches von 64 ist mit $512 \leq L \leq 1024$;
- eine Primzahl q mit einer Länge von 160 Bits (d. h. mit $2^{159} < q < 2^{160}$), für die gilt $q \mid p - 1$;
- eine Zahl g mit $0 < g < q$, die erzeugendes Element der q-elementigen Untergruppe von $\left({}^{\mathbb{Z}}/_{p\,\mathbb{Z}}\right)^{\times}$ ist.

Als *geheimer Schlüssel* eines Benutzers tritt eine Zufallszahl (Pseudozufallszahl) x mit $0 < x < q$ hinzu. Der zugehörige *öffentliche Schlüssel* ist die Zahl $y := g^x \bmod p$. Zur Signaturerzeugung benötigt man die Zahlen p, q, g und x. Zum Überprüfen von Signaturen benötigt man die Zahlen p, q, g und y.

Signaturerzeugung: Für jede zu erzeugende Signatur auf eine Nachricht (d. h. einen Bitstring) M berechnet der Signierer mit dem Secure Hash Algorithm ([SHA-1], dem Nachfolger von [SHA]) deren Hash $H(M)$ – einen 160-Bit-Wert – und erzeugt eine frische Zufallszahl (Pseudozufallszahl) k mit $0 < k < q$. Die Signatur auf die Nachricht M ist dann das Paar (r, s) von Zahlen mit

$$r := (g^k \bmod p) \bmod q,$$
$$s := \left(k^{-1}\big(H(M) + xr\big)\right) \bmod q.$$

Hierbei bezeichnet k^{-1} das mod-q-Inverse von k, also die ganze Zahl mit $k^{-1}k \equiv 1 \pmod q$ und $0 < k^{-1} < q$.

Falls $r = 0$ oder $s = 0$ gilt, kann die Signatur nicht verwendet werden; dann muß die Signaturberechnung mit einem neuen Wert k neubegonnen werden. Dieser Fall ist jedoch extrem unwahrscheinlich (vergleichbar dem Erraten des geheimen Signierschlüssels) und für die Praxis deshalb nicht relevant.

Wenn wir die Parameter p, q und g als fest vorgegeben ansehen, können wir $\mathrm{DSA}_{x,k}(M)$ schreiben für das wie oben berechnete Paar (r, s).

Signaturüberprüfung: Um anhand der öffentlichen Parametern p, q, g und y eine Signatur (r, s) auf eine Nachricht M zu überprüfen, testet der Prüfer zunächst, ob $0 < r < q$ und $0 < s < q$ ist. Falls dies nicht der Fall ist, wird die Signatur nicht akzeptiert. Andernfalls berechnet der Prüfer

$$w := s^{-1} \bmod q,$$
$$v := \left(\left(g^{(H(M)\cdot w) \bmod q} \cdot y^{(rw) \bmod q}\right) \bmod p\right) \bmod q,$$

und akzeptiert die Signatur, wenn dann $v = r$ ist.

Wir schreiben $\mathrm{DSA?}_y(M, r, s)$ für das Prädikat, das hinsichtlich fest vorgegebener Parameter p, q und g angibt, ob eine Signatur nach diesem Verfahren akzeptiert wird ($\mathrm{DSA?}_y(M, r, s) = \mathsf{wahr}$) oder nicht ($\mathrm{DSA?}_y(M, r, s) = \mathsf{falsch}$). Man kann leicht zeigen [DSA], daß in jedem Fall $\mathrm{DSA?}_y\big(M, \mathrm{DSA}_{x,k}(M)\big) = \mathsf{wahr}$ gilt.

3 Simmons' Protokoll zum Vermeiden von verdeckten Kanälen in DSA

Die im DSA benutzten Zufallszahlen k schaffen einen *verdeckten Kanal (covert channel):* Wenn der Signierer A sie nicht völlig zufällig wählt, sondern dabei auf gewisse nachprüfbare Eigenschaften der Signaturen (r, s) abzielt, kann er auf diesem Weg Information verschicken – selbst dann, wenn ihm die zu unterschreibenden Nachrichten M exakt vorgegeben werden.

[Simmons 1993] beschreibt ein interaktives Protokoll, mit dem ein »Wächter« B diesen verdeckten Kanal schließen kann.[3]

Zunächst überprüft B die öffentlichen Parameter p, q und g daraufhin, daß p und q tatsächlich Primzahlen sind mit $q \mid p-1$ und daß g in $\left(\mathbb{Z}/_{p\,\mathbb{Z}}\right)^{\times}$ tatsächlich ein Element der Ordnung q ist. (Letzteres ist gleichbedeutend damit, daß $g^q \equiv 1 \pmod{p}$, aber $g \not\equiv 1 \pmod{p}$ ist, und kann so leicht nachgeprüft werden.)

Dann wird jedesmal, wenn eine Signatur auf eine Nachricht M berechnet werden soll (wobei wir annehmen, daß die Nachricht beiden Seiten bekannt ist), folgendes Protokoll durchgeführt:

1. Der Signierer A wählt eine zu q teilerfremde Zahl k', berechnet $t := g^{k'} \bmod p$ und sendet t an den Wächter B. (k' sollte eine Zufallszahl sein mit $0 < k' < q$, dies ist jedoch für B nicht nachprüfbar.)
2. B überprüft, daß t in $\left(\mathbb{Z}/_{p\,\mathbb{Z}}\right)^{\times}$ ein Element der Ordnung q ist.
3. B wählt eine Zufallszahl k'' mit $0 < k'' < q$ und sendet diese an A.
4. A überprüft, daß $0 < k'' < q$ ist, und berechnet $k := k'k'' \bmod q$.
5. A berechnet die Signatur $(r, s) := \mathrm{DSA}_{x,k}(M)$ und schickt sie an B.
6. B überprüft das Paar (r, s) daraufhin, ob in der Tat $r = \left((t)^{k''} \bmod p\right) \bmod q$ gilt und ob $\mathrm{DSA?}_y(M, r, s) = \mathsf{wahr}$ ist. Wenn beide Bedingungen erfüllt sind, kann B die Signatur (r, s) an Dritte weitergeben.

Falls bei der Durchführung dieses Protokolles eine der angegebenen Überprüfungen fehlschlägt, hat sich eine der Seiten falsch verhalten, oder es liegt ein Datenübertragungsfehler vor. Der jeweilige Protokolldurchlauf kann dann nicht fortgesetzt werden, sondern muß von neuem begonnen werden; hierbei sollten beide Seiten die gleichen Zufallszahlen erneut einsetzen, weil B nur dann sicherstellen kann, daß A nicht doch durch selektive Kooperationsverweigerung einen verdeckten Kanal realisiert.

Bei diesem Vorgehen wirkt B zwar an der Festlegung von k mit, kann jedoch k (auch wenn k'' entgegen der Protokollvorschrift nicht völlig zufällig gewählt wird) selbst nicht berechnen [Simmons 1995].

4 Überwachung der Signierkarte

Wir zeigen, wie der Karteninhaber so mit der Signierkarte zusammenwirken und diese dabei überwachen kann, daß ein regulärer öffentlicher DSA-

[3] Siehe auch [Desmedt 1996], wo angemerkt wird, daß mit der Entscheidungsmöglichkeit des Signierers, entweder zu kooperieren oder durch Nichtkooperation das Entstehen einer Signatur zu verhindern, trotzdem ein gewisser Kommunikationskanal zur Verfügung steht.

Schlüssel und reguläre DSA-Signaturen entstehen.

Zur Überwachung der Signierkarte durch den Karteninhaber sind insgesamt drei verschiedene Punkte zu betrachten:

- Die Erzeugung der allgemeinen Parameter p, q und g;
- die Erzeugung des geheimen Signierschlüssels x;
- die Erzeugung von Signaturen zu Nachrichten M.

Der erste Punkt ist bereits im Anhang 2 von [DSA] behandelt: Ein definiertes Verfahren generiert nach Vorgabe eines Startwertes *(seed)* die Primzahlen p und q. Wird der Startwert veröffentlicht, ist leicht nachzuprüfen, daß die Primzahlen nicht speziell mit einer »Falltür« konstruiert wurden, die die Sicherheit des darauf aufbauenden Signierschlüssels in Frage stellen könnte. – Wie das die q-elementige Untergruppe von $\left(\mathbb{Z}/_{p\,\mathbb{Z}}\right)^{\times}$ erzeugende Element g gewählt wird, spielt für die Sicherheit keine Rolle.

Den dritten Punkt, die Signaturerzeugung, haben wir bereits in Abschnitt 3 angegangen: Indem für jede Signatur das Verfahren von Simmons (mit dem Karteninhaber als »Wächter«) benutzt wird, hat die Signierkarte keinerlei Möglichkeit, durch einen verdeckten Kanal etwa den geheimen Schlüssel x preiszugeben. Darüberhinaus wird der Karteninhaber – vorausgesetzt, er benutzt zur Erzeugung der Werte k'' im Protokoll aus Abschnitt 3 einen guten Zufallszahlengenerator – vor einem eventuell mangelhaften Zufallszahlengenerator der Karte [BGM 1997] geschützt: Wenn k'' gleichmäßig verteilt ist mit $0 < k'' < q$, ist dann nämlich (unabhängig von der Wahl von k') auch $k := k'k'' \bmod q$ gleichmäßig verteilt mit $0 < k < q$.

Es bleibt der Hauptpunkt, die **Schlüsselerzeugung**. Hierbei bauen wir auf dem Protokoll von Simmons auf. Wir gehen davon aus, daß die Karte beim Kartenausgeber schon einen DSA-Schlüssel x' erhalten hat; der zugehörige öffentliche Schlüssel $y' := g^{x'} \bmod p$ sei der Kartenausgabestelle und dem Karteninhaber bekannt. (x' dient als Signierschlüssel der *Karte,* nicht des Karten*inhabers.)* Sobald er die Karte A erhalten hat, führt der Karteninhaber B in Zusammenwirkung mit der Karte folgendes Protokoll[4] durch, um seinen Signierschlüssel x zu erzeugen:

1. B überprüft, daß y' in $\left(\mathbb{Z}/_{p\,\mathbb{Z}}\right)^{\times}$ ein Element der Ordnung q ist. (Wenn dies nicht der Fall ist, wurde y' fehlerhaft berechnet, und die Karte kann nicht verwendet werden.)
2. B wählt eine Zufallszahl x'' mit $0 < x'' < q$ und sendet diese an A.

[4] Auf die während des Protokolles nötige Authentisierung des legitimen Karteninhabers gegenüber der Signierkarte – genau wie bei der üblichen Verwendung der Karte zum Signieren – gehen wir bei der Darstellung nicht ein.

3. A überprüft, daß wirklich $0 < x'' < q$ gilt, und setzt dann $x := x'x'' \bmod q$. Dieser Wert wird als zukünftiger geheimer Signierschlüssel dauerhaft in der Karte gespeichert.

4. A berechnet mit $y := g^x \bmod p$ (oder äquivalent $y = (y')^{x''} \bmod p$) den öffentlichen Teil des neuen Schlüssels.

5. A erzeugt mit dem alten Schlüssel x' eine Signatur $DSA_{x',k}(y)$ für den neuen öffentlichen Schlüssel y. Der Signiervorgang (also insbesondere die Festlegung von k) wird von B mit dem Protokoll aus Abschnitt 3 »überwacht«; B überprüft dabei auch, ob wirklich eine Signatur für $(y')^{x''} \bmod p$ erzeugt wird.

6. Die Signatur $DSA_{x',k}(y)$ wird über B an den Kartenausgeber weitergereicht. Dieser hat damit eine Bestätigung, daß die an B ausgegebene Karte A einen geheimen Schlüssel enthält, zu dem der öffentliche Schlüssel y gehört. B erhält deshalb auf Antrag vom Kartenausgeber ein Zertifikat für seinen neuerzeugten Schlüssel y.

Ähnlich wie für das k im Protokoll in Abschnitt 3 gilt hier, daß $x = x'x'' \bmod q$ gleichmäßig verteilt ist im Intervall $0 < x < q$, vorausgesetzt, daß x'' gemäß einer gleichmäßigen Verteilung zufällig erzeugt worden ist. Es spielt also für den Karteninhaber im Zweifel keine Rolle, wie der ursprüngliche Schlüssel x' gewählt war. – Daß der Karteninhaber (die Protokollvorschrift verletztend) x'' in Abhängigkeit von x wählen kann, ermöglicht ihm nicht etwa, seinen geheimen Schlüssel x in Kenntnis zu bringen: Die Lage ist analog zu der beim gemeinsamen Erzeugen von k durch das Protokoll von Simmons (Abschnitt 3).

5 Einordnung

In Abschnitt 3 von [FJPP 1995] wird eine ähnliche Zielsetzung wie in diesem Beitrag verfolgt, wenn für Signaturschlüssel »eine Synthese der Standpunkte alleinige Erzeugung in Zentralen und alleinige Erzeugung unter Kontrolle der Teilnehmer« vorgeschlagen wird. Als Realisierungsmöglichkeit wird insbesondere angegeben das Zusammensetzen mehrerer mit verschiedenen Schlüsseln berechneter digitaler Signaturen zu einer Gesamtheit, die nur dann als gültig angesehen werden soll, wenn jede Einzelsignatur gültig ist. (Die Einzelsignaturen können in separaten Geräten berechnet werden.) Speziell können dabei zwei getrennte Signierschlüssel verwendet werden, von denen einer zentral und einer vom Nutzer generiert worden ist.

Diese Herangehensweise, Signaturen zu kombinieren, unterscheidet sich von dem im vorliegenden Beitrag dargestellten Ansatz außer durch das starke Anwachsen der Signaturlänge wesentlich in einem Punkt: Man

hat es dabei mit (mindestens) zwei selbständigen Signiereinheiten zu tun, von denen jede einen Geheimschlüssel zu speichern und dabei dessen Auslesen zu verhindern hat. Das Signiergerät, das mit dem vom Benutzer erzeugten Schlüssel umzugehen hat, sollte also einerseits über Sicherheitsmaßnahmen verfügen, die die Schlüsselgeheimhaltung gewährleisten können; andererseits soll seine Funktionsweise soweit nachvollziehbar sein, daß es für den Schlüsselinhaber als vertrauenswürdig gelten kann. Bei dem hier in Abschnitt 4 vorgestellten Verfahren zum gemeinsamen Erzeugen eines DSA-Schlüssels sowie bei Simmons' Methode zum Vermeiden von verdeckten Kanälen bei DSA gelten in dieser Hinsicht schwächere Anforderungen: Die vom Karteninhaber zusätzlich zur Signierkarte eingesetzte »Kartenumgebung« – das System, das für ihn »seine« Schritte der Protokolle durchführt –, muß nur kurzzeitig Werte geheimhalten, und das langfristige Speichern des Geheimschlüssels bleibt alleine der Signierkarte überlassen. Letztere kann mit den entsprechenden Schutzvorrichtungen ausgerüstet werden, während die Kartenumgebung daraufhin entworfen werden kann, daß ihre Funktionsweise nachvollziehbar und überprüfbar wird. Nachteilig ist dabei allerdings, daß die langfristige Sicherung des geheimen Signierschlüssels vor Auslesen nach Verlust der Karte alleine dem Produkt eines einzigen Herstellers überlassen bleiben würde – nach dem Abhandenkommen der Signierkarte ließe sich der Grundsatz der Überwachung der Karte zwangsläufig nicht mehr aufrechterhalten. Insofern sind die beiden Ansätze als komplementär anzusehen: Beim Zusammensetzen von Signaturen gemäß [FJPP 1995] können Einzelsignaturen auf den in diesem Beitrag dargestellten Methoden beruhen.

Literatur

[AK 1996] R. J. ANDERSON, M. G. KUHN: Tamper Resistance – a Cautionary Note. – *Proceedings of the Second USENIX Workshop on Electronic Commerce.* USENIX Association, 1996. 1–11. [ISBN 1-88044-683-9]

[BGH 1998] BUNDESGERICHTSHOF: Urteil vom 17. 6. 1997 – X ZR 119/94 (Nürnberg). – Abgedruckt in *NJW* **1998**, 79–81. [ISSN 0341-1915]

[BGM 1997] M. BELLARE, S. GOLDWASSER, D. MICCIANCIO: "Pseudo-Random" Number Generation Within Crypgraphic Algorithms: The DSS Case. – B. S. Kaliski, ed.: *Advances in Cryptology – CRYPTO '97.* Lecture Notes in Computer Science **1294**. Springer-Verlag, 1997. 277–291. [ISBN 3-540-63384-7]

[Desmedt 1996] Y. DESMEDT: Simmons' Protocol is Not Free of Subliminal Channels. – *9th IEEE Computer Security Foundations Workshop.* 1996. 170–175. [ISBN 0-8186-7522-5]

[DSA] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST): *Digital Signature Standard (DSS).* FIPS PUB **186**. 1994 May 19.

[FJPP 1995] H. FEDERRATH, A. JERICHOW, A. PFITZMANN, B. PFITZMANN: Mehrseitig sichere Schlüsselerzeugung. – P. Horster (Hrsg.): *Trust Center.* DuD-Fachbeiträge. Vieweg, 1995. 117–131. [ISBN 3-528-05523-5]

[GUQ 1992] L. C. GUILLOU, M. UGON, J. J. QUISQUATER: The Smart Card: A Standardized Security Device Dedicated to Public Cryptology. – G. J. Simmons, ed.: *Contemporary Cryptology. The Science of Information Integrity.* IEEE Press, 1992. Chapter 12, 561–613. [ISBN 0-87942-277-7]

[MvOV 1997] A. J. MENEZES, P. C. VAN OORSCHOT, S. A. VANSTONE: *Handbook of Applied Cryptography.* CRC Press, 1997. [ISBN 0-8493-8523-7]

[Petri 1997] T. B. PETRI: Anscheinsbeweis. – *Datenschutz und Datensicherheit* **21** (1997) 11. [ISSN 0724-4371]

[PP 1997] T. P. PEDERSEN, B. PFITZMANN: Fail-stop signatures. – *SIAM Journal on Computing* **26** (1997) 291–330. [ISSN 0097-5397]

[Rüßmann 1998] H. RÜSSMANN: Haftungsfragen und Risikoverteilung bei ec-Kartenmißbrauch. – *Datenschutz und Datensicherheit* **22** (1998) 395–400. [ISSN 0724-4371]

[SHA] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST): *Secure Hash Standard.* FIPS PUB **180**. 1993 May 11.

[SHA-1] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST): *Secure Hash Standard.* FIPS PUB **180-1**. 1995 April 17.

[Simmons 1993] G. J. SIMMONS: An Introduction to the Mathematics of Trust in Security Protocols. – *The Computer Security Foundations Workshop VI.* IEEE Computer Society Press, 1993. 121–127. [ISBN 0-8186-3950-4]

[Simmons 1995] G. J. SIMMONS: Protocols that ensure fairness. – *Codes and Cyphers.* Proceedings of the Fourth IMA Conference on

Cryptography and Coding, December 1993. Southend-on-Sea, Essex, Formara Limited, 1995. 383–394. [ISBN 0-905091-03-5]

[YY 1996] A. YOUNG, M. YUNG: The Dark Side of "Black-Box" Cryptography or: Should We Trust Capstone? – N. Koblitz, ed.: *Advances in Cryptology – CRYPTO '96.* Lecture Notes in Computer Science **1109**. Springer-Verlag, 1996. 89–103. [ISBN 3-540-61512-1]

[YY 1997] A. YOUNG, M. YUNG: Kleptography: Using Cryptography Against Cryptography. – W. Fumy, ed.: *Advances in Cryptology – EUROCRYPT '97.* Lecture Notes in Computer Science **1233**. Springer-Verlag, 1997. 62–74. [ISBN 3-540-62975-0]

Dieser Beitrag zur Konferenz »Sicherheitsinfrastrukturen« (Hamburg-Harburg, 9. und 10. März 1999) erschien in abweichender Formatierung in: P. Horster (Hrsg.): *Sicherheitsinfrastrukturen.* DuD-Fachbeiträge. Vieweg, 1999. 238–246. [ISBN 3-528-05709-2]