Sicherheitsanalyse von Betriebssystemen für Mobile Endgeräte

Tobias Murmann, Heiko Rossnagel

Lehrstuhl für M-Commerce und Mehrseitige Sicherheit
Johann Wolfgang Goethe Universität Frankfurt
Gräfstr. 78
60054 Frankfurt
TMurmann@t-online.de
heiko.rossnagel@m-lehrstuhl.de

Abstract: Es gibt einige Initiativen mobile Endgeräte als "Trusted Pocket Signers" einzusetzen um elektronische Signaturen zu erzeugen. Die eigentliche Signatur wird dabei mit Hilfe einer konventionellen Signaturkarte erzeugt. Das mobile Endgerät dient als Kartenleser, Speichermedium für das zu signierende Dokument und als Display für die Signaturapplikation. Daher hat das auf dem mobilen Endgerät genutzte Betriebssystem eine entscheidende Bedeutung bei der Sicherstellung der Integrität und Zurechenbarkeit der elektronischen Signatur. Weiterhin werden mobile Endgeräte eingesetzt um Außendienstmitarbeitern einen Zugang zum Backend der Firma zu ermöglichen. Im Rahmen dieses Beitrags haben wir die Sicherheitseigenschaften aktuell verfügbarer mobiler Betriebssysteme untersucht und festgestellt, dass kein einziges davon eine ausreichende Sicherheit zur Signaturerstellung bietet. Auch für den Zugriff auf ein Firmenbackend sind die Sicherheitseigenschaften oft nicht ausreichend. Um dieses Problem zu addressieren stellen wir zwei Möglichkeiten vor, wie die Sicherheit mobiler Endgeräte verbessert werden kann.

1 Einleitung

Mobile Endgeräte werden immer leistungsfähiger und können sich anhand von neuen Funktionalitäten ein breiteres Spektrum von Einsatzfeldern im betrieblichen Umfeld erschließen. Personal Digital Assistents (PDA's) und Smartphones als mobile Endgeräte ermöglichen Nutzern zeit-, und ortsunabhängigen Zugriff auf sensible, personengebundene Daten, wodurch eine Produktivitätssteigerung ermöglicht wird. Für den Einsatz mobiler Endgeräte mit sensiblen Daten muss jedoch deren Sicherheit gewährleistet werden. Sensible Daten können beispielsweise Patientendaten, Kundenlisten oder Adressdaten sein.

Unternehmen nutzen mobile Endgeräte um ihren Außendienstmitarbeitern Zugriff auf ihr Backend zu ermöglichen. Da die Unternehmensdaten sehr vertraulich sein können,

muss der Zugang zu diesen Daten entsprechend sicher gestaltet sein. Ziel des durch die Europäische Union geförderten Projekts "Wireless Trust for Mobile Business (WiTness)" [Wi04] war es, solch einen sicheren Zugriff auf das Backend mittels GSM Technologie zu ermöglichen. Abbildung 1 zeigt ein Anwendungsszenario bei dem ein "pervasive salesman" sicheren und vom Unternehmen kontrollierten Zugang auf alle ihm, im Informationssystem des Unternehmen, zugänglichen Daten hat. Der Zugang wird durch ein Sicherheitsmodul kontrolliert, das auf einer SIM mit zusätzlicher Sicherheitsfunktionalität basiert.

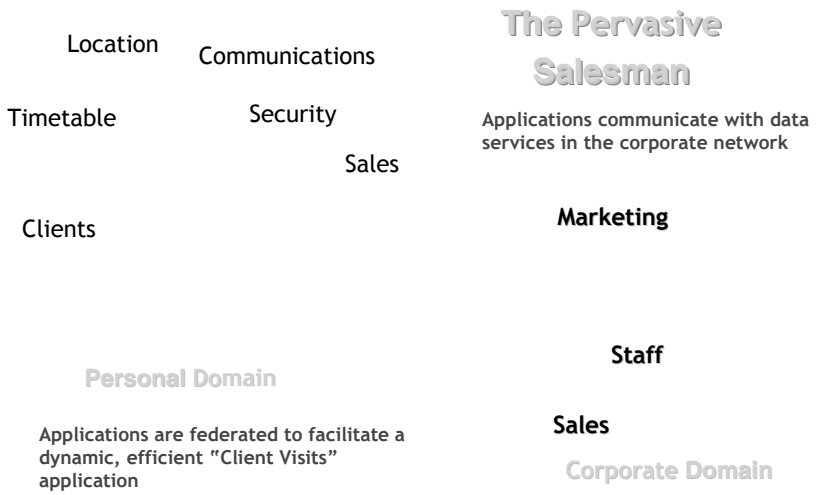

Abbildung 1: WiTness Pervasive Salesman Szenario [Wi04]

Aber auch wenn die Kommunikation und der Zugang zum Backend gesichert sind, ist das mobile Endgerät selbst nicht vor möglichen Angriffen geschützt. Falls Unternehmensdaten auf dem Endgerät gespeichert werden, könnte ein Angreifer versuchen den Zugangskontrollmechanismus des Endgerätes zu umgehen, um Zugriff auf die gespeicherten Daten zu erhalten.

Weiterhin existieren zahlreiche Initiativen, die mobile Endgeräte als sogenannte „Trusted Pocket Signer" zur Erstellung elektronischer Signaturen nutzen [Mo04]. Hierbei wird eine herkömmliche Signaturkarte (nach SigG [Si01]) verwendet, um die eigentliche Signatur zu erstellen. Das mobile Endgerät fungiert als Kartenleser, Speicherort für das zu signierende Dokument sowie als Display für die Signaturrapplikation. Hierbei muss sichergestellt werden, dass die auf dem Display angezeigten Daten auch diejenigen sind, die mittels der Signaturkarte unterschrieben werden (WYSIWYS[1]). Daher kommt dem auf dem mobilen Endgerät eingesetzten Betriebssystem eine zentrale Bedeutung für die Sicherstellung der Integrität und Zurechenbarkeit der digitalen Signatur zu.

---

[1] What You See Is What You Sign

Falls ein Betriebssystem einen Mangel bei Rechtevergabe, Speicherschutz, Prozesserzeugung und -trennung oder Schutz der Dateien aufweist, kann ein Angreifer Zugriff auf die einzelnen internen Prozesse erlangen. Dies könnte er nutzen, um gefälschte elektronische Signaturen zu erzeugen.

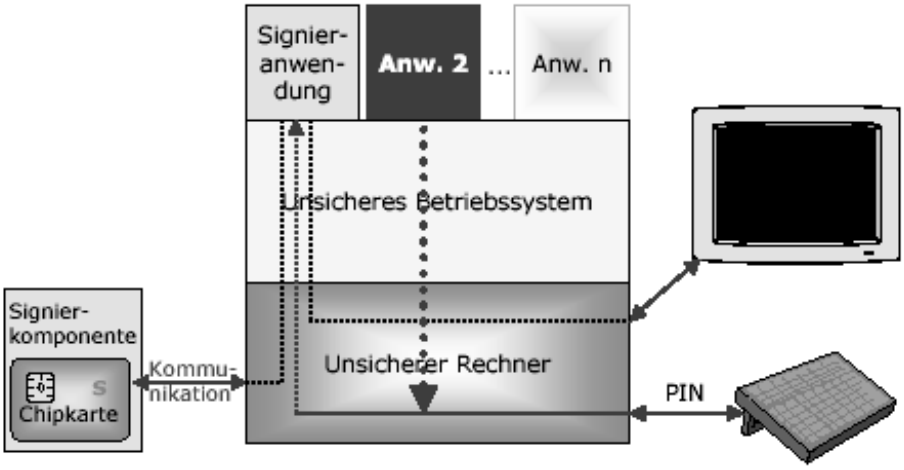

Abbildung 2: Manipulierte digitale Signatur [Fe03]

Abbildung 2 verdeutlicht, dass Anwendung 2 als schädliches Programm beispielsweise die PIN abfangen kann Noch deutlich gefährlicher ist es jedoch, falls die gefährliche Anwendung die zu signierenden Daten zwischen Kontrolle des Nutzers am Display und dem eigentlichen Signieren ändert. Ein schädliches Programm kann somit auf Grund des praktisch ungehinderten Hardwarezugriffs alle Daten, welche zur Signaturanwendung zum Signieren übermittelt werden, manipulieren.

Daher werden in Abschnitt 2 die aktuell am Markt verfügbaren mobilen Betriebssysteme überprüft und einige wesentliche Sicherheitsmängel vorgestellt. Diese Sicherheitsanalyse ist notwendig, um eine Beurteilung der Möglichkeit sicherer Signaturen zu ermöglichen. Abschnitt 3 soll einen Ausblick präsentieren, wie diese Problematik mittels Software oder Hardware-Lösungen gelöst werden könnte. Der abschließende Abschnitt 4 fasst die gewonnen Ergebnisse zusammen.

2 Sicherheitsanalyse aktueller mobiler Betriebssysteme

2.1 Pocket PC 2003

Microsoft Pocket PC [Po04] bietet keine Möglichkeit, Daten auf dem mobilen Endgerät zu verschlüsseln. Auch die interne Kommunikation ist nicht gesichert. Pocket PC grenzt weder Speicherblöcke noch Applikationen wirkungsvoll von einander ab. Jede Applikation kann ihre Prioritäten anpassen, andere Applikationen beenden, auf deren Speicherbereiche zugreifen oder auch den Wechsel in den Stromsparmodus unterbinden.

Passwörter können durch den Nutzer deaktiviert werden und sind oft in der Standardeinstellung bereits deaktiviert. Auch kann externer Speicher, beispielsweise eine SD-Karte, durch einen Angreifer einfach aus dem Endgerät genommen werden und so können die darauf gespeicherten Daten gestohlen werden.

Gefälschte Dialoge sind möglich, da das Microsoft Betriebssystem Active X und Java unterstützt. Mit deren Hilfe kann ein gefälschter Dialog aufgebaut werden.

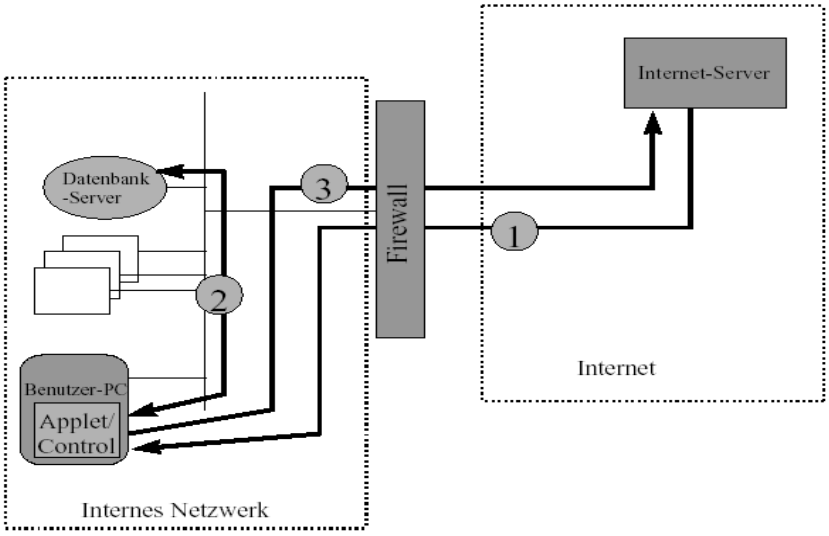

Abbildung 3: Angriffszenario mobiler Code [FH91]

1. Der Benutzer lädt das Applet (Java) bzw. Control (ActiveX) von einem Webserver und es wird auf dem mobilen Endgerät des Kunden ausgeführt.
2. Das Applet/Control nutzt die Rechte des Besitzers des mobilen Endgerätes, um Zugriff auf die Firmendatenbank zu erhalten und kopiert Daten auf das mobile Endgerät.
3. Das Applet/Control sendet die gewonnenen Daten an den Internet - Server zurück.

Dabei existiert für ein Java-Applet die so genannte Sandbox, welche die Rechte des Applets auf die Hardware und Software beschränkt. Allerdings können dem Applet durch den Nutzer zu viele Rechte eingeräumt werden, oder der Angreifer nutzt eine der vielen Sicherheitslücken in der Java Virtual Machine aus. Zusätzlich existiert als Absicherung für Nutzer ein Code Signing Mechanismus, mit dem der Ursprung von Programmen zertifiziert werden kann. Anhand dieses Verfahrens kann nur die Urheberschaft eines Programms bestimmt werden, der eigentliche Inhalt kann jedoch schädlich sein. Da aber Pocket PC keine Verwaltung von Zertifikaten ermöglicht, muss jede Form von mobilem Code in den Einstellungen deaktiviert werden.

Die Existenz versteckter Hintertüren ist bei Pocket PC theoretisch möglich, da der Quellcode nicht offen liegt. Ein Schutz vor Buffer – Overflows kann auch durch zusätzliche Software nicht erreicht werden, genauso wie einen wirksamen Schutz vor der Manipulation der DMA – Funktionalität. Manipulierte Programme haben noch immer die Möglichkeit mit allen Rechten des Benutzers zu agieren, da keine Rechtevergabe existiert. Pocket PC 2003 besitzt eine Vielzahl an Sicherheitslücken, die auch nicht durch zusätzliche Sicherheitssoftware wie beispielsweise PDA-Secure [PS04] oder PDA-Defense [PD04] vollständig geschlossen werden können. Somit existieren Sicherheitsrisiken, die es nicht erlauben Pocket PC als Betriebssystem für einen„trusted pocket signer“ einzusetzen. Auch bei dem Anwendungsbeispiel des „WiTness Pervasive Salesman“ in Abbildung 1, sollte PocketPC nicht benutzt werden.

2.2 PalmOS 5.0

Wie beim Pocket PC fehlen bei PalmOS [Pa04] eine wirksame Rechtevergabe und eine Trennung der Prozesse. Es existiert kein sicherer Pfad zwischen Anwendungen und Kernel und die Kommunikation ist angreifbar. Weiterhin kann der Benutzer, wie bei allen Betriebssystemen, nicht prüfen, ob der Status des Endgerätes sicher ist. Dies könnte zum Beispiel durch eine LED ermöglicht werden, die nach einer Überprüfung des Status des PDA farblich anzeigt, ob dieser sich in einem sicheren Stadium befindet. Darauf wird jedoch in Kapitel 3 näher eingegangen.

Kommt ein Angreifer in den Besitz des aktivierten Endgerätes so kann er dieses mit jedem beliebigen PC synchronisieren und Mallware installieren. Auch Passwörter sind bei Palm OS nur mangelhaft geschützt. Da auch bei Palm OS der Quellcode nicht offen liegt, existiert die Möglichkeit von versteckten Hintertüren. Mobiler Code kann ebenfalls über Java auf dem mobilen Endgerät ausgeführt werden, so dass das Angriffsszenario „mobiler Code“ wie in Abbildung 2 relevant ist. Palm unterstützt auch kein Zertifikatsmanagment, sodass ein manipuliertes Zertifikat nicht erkennbar wäre.

Direct Memory Access wird auch von Palm OS durch die Unterstützung von ARM- und DragonBall- Prozessoren unterstützt

Dies zeigt, dass PalmOS, genauso wie PocketPC, für das „Pervasive Salesman“ Verfahren nicht sicher genug ist. Ein manipuliertes Programm hat die Möglichkeit mit allen Rechten des Nutzers zu agieren. Wie auch bei Pocket PC existiert unter anderen die Sicherheitssoftware PDA-Secure [PS04] oder PDA-Defense [PD04]. Selbst mit deren Einsatz existieren noch Sicherheitsrisiken, die es nicht erlauben, Palm OS als sicheres Betriebssystem für elektronische Signaturen nach SigG [Si01] einzusetzen. Die aktuelle Betriebssystemversion 6.1 von PalmOS verspricht Speicherschutz und Dateischutz, allerdings sind noch keine mobilen Endgeräte für diese Version verfügbar.

2.3 Symbian 7.0

Symbian [Sy04] bietet einen besseren Schutz als Palm OS und Pocket PC 2003. Das Endgerät kann im Firmennetzwerk anhand von Access Control Listen administriert werden. Mit deren Hilfe können bestimmte Inhalte vor dem Zugriff von anderen Device

Management Servern geschützt werden, so dass die Daten nur mit einem bestimmten Server synchronisiert werden können.

Bisher sind keine größeren Sicherheitslücken im Betriebssystem bekannt. Jedoch existiert das Nokia Wartungsprogramm Wintesla [UC03]. Dieses ermöglicht weitgehende Eingriffe in das mobile Endgerät, auch wenn es gesperrt ist. Der Angreifer erhält vollen Zugriff auf alle Einstellmöglichkeiten des Endgerätes, kann es mit dem gewonnen Wissen entsperren und hat vollen Zugriff auf die gespeicherten Daten, was jegliche Sicherheitsansprüche von Nokia - Endgeräten ad absurdum führt.

Mobiler Code und damit auch gefälschte Dialoge sind durch die Unterstützung von Java möglich. Im Gegensatz zu Palm OS und Pocket PC ist jedoch zum Schutz vor gefälschten Zertifikaten ein Zertifikatsmanagment installiert. Weiterhin kann der Benutzer den Sicherheitsstatus des Endgeräts nicht prüfen und hat auch kaum Möglichkeiten, zusätzliche Sicherheitssoftware auf dem Endgerät zu installieren.

Daher ist eine Eignung von Symbian Endgeräten zum Erstellen qualifizierter Signaturen nicht gegeben da noch Sicherheitslücken existent sind, wie beispielsweise eine mangelnde Prozesstrennung und vor allem Tools existieren, die jegliche Sicherheitseinrichtungen in Symbian aushebeln können.

Ohne das Problem des Wintesla Tool, bietet Symbian mehr Sicherheitsmerkmale als PalmOS oder PocketPC. Mit dem Merkmal der Zugangskontrollliste und der Unterstützung eines Zertifizierungsmanagements eignet sich Symbian für das Anwendungsbeispiel des „Pervasive Salesman“. Aber auch mit besserem Schutz der gespeicherten Daten ist Symbian bei weitem noch nicht sicher genug für das “Trusted Pocket Signer” Anwendungsszenario.

2.4 Linux

Das Linux – Betriebssystem eröffnet dem Benutzer die meisten Sicherheitsfunktionen der bisher vorgestellten Betriebssysteme. Aufgrund der Möglichkeit, Berechtigungen für Prozesse, Daten usw. festzulegen, ist ein besserer Schutz vor dem Missbrauch der Daten gewährleistet. Allerdings existieren weiterhin eine Vielzahl an Sicherheitslücken, wie beispielsweise die DMA – Funktionalität, die manuell deaktiviert werden muss, oder die Möglichkeit, eine Synchronisation ohne Authentifizierung durchführen zu können. Weiterhin existieren Viren und Würmer, wenn auch nicht direkt für die mobilen Linux - Distributionen. Es verdeutlicht aber, dass auch diese Bereiche gefährdet sind und der Schutz anhand von Berechtigungen nicht ausreichend ist, wie auch beispielsweise der Schutz vor Buffer – Overflows. Zusätzlich existiert praktisch keine Zusatzsoftware für Linux- betriebene Endgeräte, sodass kein zusätzlicher Schutz eingerichtet werden kann. Auch existieren zu wenige Linux Endgeräte und ein Umstellen der Endgeräte vom bereits installierten Betriebssystem auf Linux ist aufwendig und nicht risikofrei.

Die SUSE-Distribution wurde in Kombination mit IBM-Server durch das Bundesamt für Sicherheit in der Informationstechnik (BSI) nur mit der Stufe EAL2 zertifiziert. [BS03]

Somit könnten PDA's mit einer auf dem Standard-Kernel aufbauende Distribution keine Signaturen nach SigG [Si01] erstellen, die der handschriftlichen Unterschrift rechtlich gleichgestellt wären.

Linux bietet die besten Sicherheitsmerkmale für mobile Endgeräte. Mit der Realisierung von vielen Sicherheitsmerkmalen ist Linux das sicherste gängige mobile Betriebssystem und ermöglicht das „Pervasive Salesman" Szenario.

3 Möglichkeiten zu Verbesserungen der Sicherheit

Die beiden folgenden Lösungsvorschläge befinden sich im Entwicklungsstadium und sind aktuell nur begrenzt einsetzbar. Trotzdem ermöglichen sie zukünftig einen besseren Schutz des PCs bzw. des mobilen Endgerätes. Ziel dieser Ansätze ist es, durch eine strikte Rechteverteilung auf den untersten Schichten die internen Prozesse zu schützen. Nur durch eine systemweite Trennung von Speicher, Zugriffs- und Ein-/Ausgabe – Rechten für Prozesse und Anwendungen kann ein System vor jeder Form von schädlichen Programmen geschützt werden. Da ein schädliches Programm nicht wie in bisherigen Systemen alle Rechte des Benutzers besitzt, wird durch die vorgestellten Lösungen versucht, die Schadensmöglichkeiten zu minimieren. Der Benutzer erhält vor allem das erste Mal die Möglichkeit zu prüfen, ob sich der Rechner in einem sicheren Zustand befindet bzw. ob er mit dem Kernel sicher kommuniziert. Dies ist bei aktuellen Systemen nicht möglich.

3.1 Perseus

Perseus ist ein Open Source Projekt der Uni Saarbrücken [Pe04]. Ziel ist es einen kleinen Betriebssytem-Kernel als sichere Plattform zu entwickeln. Zusätzlich soll das User Interface dem Nutzer absolut sicher anzeigen, in welchem Status das System sich befindet, ohne dass ein gefährliches Programm diese Anzeige manipulieren kann. Grundsätzlich ist ein Kernel für die Geräte-, Datei-, Speicher-, und Prozess - Verwaltung zuständig und wird direkt nach dem Bootvorgang geladen. Der Perseus - Kernel soll sicherheitskritische Anwendungen schützen, indem die einzelnen Prozesse isoliert voneinander ablaufen. Der Ansatz von Perseus ist daher, dass das eigentliche Betriebssystem wie eine Anwendung betrieben wird, und damit ist der Perseus – Kernel in der Schichtenarchitektur unterhalb des Betriebssystems angesiedelt. Nur durch die Einbettung von Perseus unter dem weiterhin nötigen Betriebssystem gelingt es Perseus, isolierte Prozesse systemweit zwischen den Anwendungen zu ermöglichen. Isolierte Prozesse sind jedoch nicht für die Anwendungen innerhalb des Standard-Betriebssystems möglich, sondern nur zwischen den einzelnen „secure applications" und dem Perseus-Betriebssystem. Abbildung 4 zeigt die Perseus System Architektur.

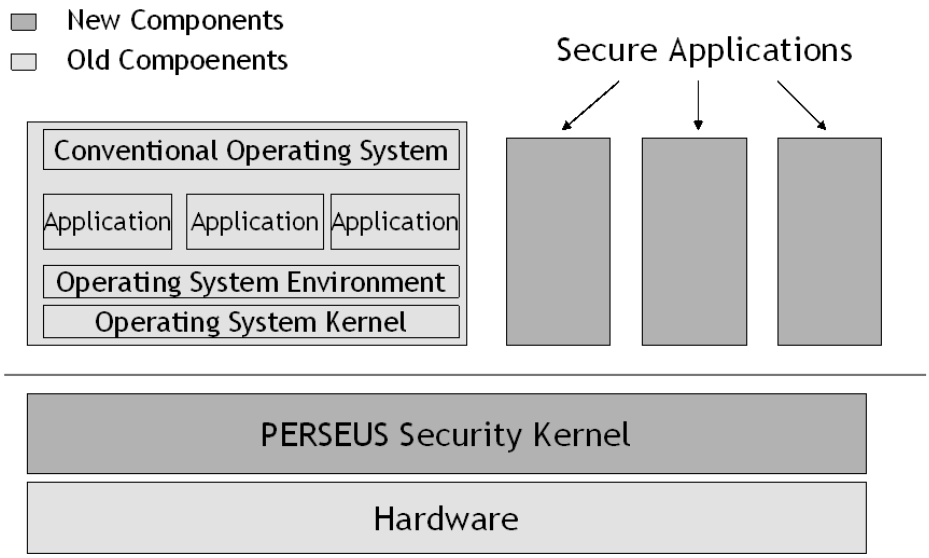

Abbildung 4: Perseus System Architektur [Pe04]

Das Trustworthy User Interface reserviert bei dem Prototypen von Perseus eine Zeile am oberen Bereich des Monitors, welche permanent unter Kontrolle des Sicherheitskernel bleibt.

Da die Zeile unter alleiniger Kontrolle von Perseus liegt, kann diese von einem kompromittierten Betriebssystem nicht missbraucht werden. Falls die Anzeige informiert, dass der Benutzer mit dem Perseus – Kernel kommuniziert, ist die Kontrolle des Displays und der Tastatur alleine beim Sicherheitskernel.

3.2 Trusted Computing

Das „Trusted Platform Modul“ (TPM) wurde von der „Trusted Computing Group (TCG)“, ehemals „Trusted Computing Platform Alliance (TCPA)” spezifiziert [TC04].

Die TCG Hardware besteht aus den zwei manipulationssicheren Modulen TPM und CRTM (Core Root of Trust for Measurement). Da ausschließlich das Betriebssystem die von der TCG spezifizierten Komponenten kontrolliert, ist bei der Nutzung die Vertrauenswürdigkeit des verwendeten Betriebssystems von entscheidender Bedeutung [GR04].

Zurzeit werden zwei Betriebssysteme entwickelt, die TCG konforme Hardware unterstützen werden. Microsoft entwickelt eine Sicherheitstechnologie mit dem Namen Next Generation Secure Computing Base (NGSCB) [Mi04], die in zukünftige Versionen von Windows integriert werden soll. Auserdem gibt es Initiativen eine Linux Distribution zu entwickeln, die TCG Sicherheitsmodule unterstützt [MS03].

Das TPM – Hardwaremodul kann als erweiterte Smartcard angesehen werden, in dem Geheimnisse in- und außerhalb des TPM erstellt und gespeichert werden können [Pe02]. Diese Geheimnisse sind symmetrische und asymmetrische Schlüssel, die dazu genutzt werden, die Vertrauenswürdigkeit von Dateien, das Signieren von Daten und die Authentifikation auf der Plattform gegenüber Dritten sicherzustellen. Weiterhin werden Hashwerte zur Identifizierung der vertrauenswürdigen Hard- und Softkomponenten überprüft und in Datenintegritätsregistern abgespeichert. Damit das TPM aktiv ist, muss es hardwaremäßig angestellt und softwaremäßig aktiviert werden.

Für jede Komponente (z.B. BIOS, OS-Loader und Betriebssystem) kann beim Systemstart ein Hashwert erstellt, der an das TPM übermittelt wird. Diese werden in den „Platform Configuration Registers“ gespeichert. Dort wird überprüft, ob die aktuell gebildeten Hashwerte mit denen auf dem TPM gespeicherten übereinstimmen. Falls dies der Fall ist, kann der Nutzer davon ausgehen, dass die Komponenten bzw. die darauf gespeicherten Daten nicht manipuliert wurden, da sich sonst der Hashwert ändert und das System bzw. die Software eine Meldung an den Nutzer ausgibt. Dadurch kann eine Authentifizierungskette gebildet werden, die mit dem CRTM beginnt [Pe02].

Das Betriebssystem kann dann einen vertrauenswürdigen Bereich (z.B. der “nexus” von NGSCB) bilden, in dem sicherheitskritische Anwendungen ausgeführt werden. Die einzelnen Anwendungen in diesem vertrauenswürdigen Bereich werden strikt voneinander getrennt ausgeführt und ein Zugriff von außen auf den trusted Bereich wird verhindert. Unzertifizierte Software bekommt keinen Zugang zu dem trusted Bereich.

4 Zusammenfassung

Aktuell verfügbare mobile Betriebssysteme eignen sich nicht, rechtlich verbindliche digitale Signaturen zu erstellen. Keines der Betriebssysteme unterstützt eine sichere Ein- und Ausgabe der Daten. Zusätzlich existiert in den Betriebssystemen noch eine Vielzahl von offenen Sicherheitslücken.

Die Lösungen durch Perseus bzw. die Spezifikationen der TCG müssen durch die Hersteller der Betriebssysteme in zukünftigen Versionen umgesetzt werden, bzw. vergleichbare entwickelt werden. Erst dann wird es mobilen Endgeräten möglich, in einem größeren Umfang als bisher auch in sicherheitskritischen Bereichen eingesetzt zu werden.

Bis dahin besteht mit dem Einsatz mobiler Endgeräte jedoch ein enormes Sicherheitsrisiko, weshalb eine genaue Abwägung bezüglich des Einsatzes nötig ist. Vor allem ist jedoch der Einsatz von Zusatzsoftware, wie z.B. PDA Defence, aktuell dringend zu empfehlen, da somit zumindest ein Teil der Sicherheitsrisiken behoben werden kann. Der Einsatz einer Vielzahl von Sicherheitssoftwareprodukten überfordert aber die durchschnittlichen Anwender, schafft zusätzliche Kosten und stellt einen hohen administrativen Aufwand dar.

Literaturverzeichnis

[BS03] Bundesamt für Sicherheit in der Informationstechnik (2003): „BSI-DSZ-CC-0216-2003" unter: www.bsi.bund.de/zertifiz/zert/reporte/0216a.pdf

[Fe03] H. Fedderath: Digitale Signatur und Public Key Infrastruktur www-sec.uni-regensburg.de/security/5PKI.pdf

[FH91] Fox, D.; Horster, P. (1999): „Datenschutz und Datensicherheit" in DuD, Verlag, Braunschweig, Seite 194

[GR04] D. Günnewig, K. Rannenberg, A.R. Sadeghi, C. Stüble: Trusted Computing Platforms – Zur technischen und industriepolitischen Situation und Vorgehensweise; Trusted Computing – Technik, Recht und gesellschaftspolitische Implikationen vertrauenswürdiger Systemumgebungen; 2004; S. 154-162

[Mi04] Microsoft, Next-Generation Secure Computing Base, www.microsoft.com/resources/ngscb/default.mspx

[Mo04] Mobile Electronic Transactions, www.mobiletransaction.org/index.html

[MS03] R. MacDonald, S. Smith, J. Marchesini, O. Wild: Bear: An Open-Source Virtual Secure Coprocessor based on TCPA, www.cs.dartmouth.edu/~sws/papers/msmw03.pdf

[Pa04] Palm Website, www.palm.com

[PD04] PDA Defense Website, www.pdadefense.com/

[Pe02] S. Pearson, et al. (2002): „Trusted Computing Platforms - TCPA Technology in context", Prentice Hall PT., New Jersey, S. 5

[Pe04] B. Pfitzmann, C. Stüble: PERSEUS: A Quick Open-Source Path to Secure Electronic Signatures, www.perseus-os.org/

[Po04] Windows Mobile – based Pocket PCs, www.microsoft.com/windowsmobile/products/pocketpc/default.mspx

[PS04] PDASecure – The encryption software, www.pdasecure.de/

[Sy04] Symbian OS – the mobile operating system, www.symbian.com

[Si01] Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften vom 16. Mai 2001, BGBl. I, S. 876

[TC04] TCG – Trusted Computing Group, www.trustedcomputing.org/home

[UC03] WinTesla v.5.31 Nokia Service Software for Windows, ucables.com/nokia/service/wintesla.html

[Wi04] European IST Project „Wireless Trust for Mobile Business" (WiTness), www.wireless-trust.org