IT-Sicherheit mobiler Endgeräte

Diplomarbeit

Abgabe am:	31. Januar 2005
Studienbereich:	Angewandte Informatik
Hochschule:	Fachhochschule Fulda
Studienrichtung:	Medieninformatik

von:	Michael Ruck Mackenrodtstraße 4b 36039 Fulda diplomarbeit@cwsnet.de
Matrikelnummer:	1 44 99 5

Referent:	Prof. Dr. Ulrich Bühler *Fachhochschule Fulda* u.buehler@informatik.fh-fulda.de
Co-Referent:	Werner Storch *EDAG Engineering & Design AG*, Fulda werner.storch@edag.de

Erklärung

Hiermit erkläre ich, dass ich diese Diplomarbeit selbständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel verwendet, sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet habe.

(Ort, Datum)

(Michael Ruck)

ANMERKUNG: Diese Diplomarbeit enthält u.a. Informationen über Geschäftsprozesse der *EDAG Engineering & Design AG*. Diese Informationen unterliegen dem Datenschutz und dürfen nur durch die Zustimmung des Verfassers bzw. eines Verantwortlichen der *EDAG Engineering & Design AG* an Dritte weiter gegeben werden. Eine Weitergabe dieser Informationen **ohne** Zustimmung der oben genannten Personen wird strafrechtliche Maßnahmen nach sich ziehen.

Danksagung

Bevor ich mit dem eigentlichen Thema beginne, ist es erst mal an der Zeit Danke zu sagen. Diese Diplomarbeit wäre ohne die Unterstützung vieler Personen, die mir mit Rat und Tat zur Seite standen, nicht möglich gewesen.

Zuerst möchte ich mich bei meinen Betreuern von der *EDAG Engineering & Design AG*, allen voran Werner Storch sehr herzlich bedanken. Er hatte immer ein offenes Ohr für meine Fragen, Probleme und Vorschläge hatte. Nicht zu vergessen sind aber auch Markus Grünkorn, Thomas Diegmüller und Peter Feuerstein und natürlich alle anderen, die mich hier unterstützt haben.

Auch meinem betreuenden Professor, Herrn Dr. Ulrich Bühler, möchte ich für seine Hilfestellung und seine Ratschläge recht herzlich danken.

Für die technische Unterstützung möchte ich mich bei Frau Klein von *Siemens Mobile* für die Bereitstellung des *Siemens SX1* und Herrn Hinz von *Motorola* für die Bereitstellung des *Motorola MPx200* bedanken. In diesem Zuge richte ich auch meinen Dank an Frau Donges von *Vodafone Deutschland*, die mir die Kontakte hergestellt hat.

Auch ein herzliches Dankeschön an Herrn Odenthal von *Pointsec*, Herrn Tewes von *CC Compunet* und Herrn Sosna von *ubitexx*, die mir die Evaluierungslizenzen für die von mir getesteten Lösungen zur Verfügung gestellt haben. Ebenso vielen Dank an Herrn Micheel-Sprenger und Herrn Wiegand von *ECOPLAN* für die Live-Präsentation von *Pylon Anywhere* und die Beantwortung meiner technischen Fragen.

Vielen Dank an die Pressestelle des *Bundesamtes für Sicherheit in der Informationstechnik*, die mir das *GSTOOL* für die Dauer meiner Diplomarbeit als Vollversion zur Verfügung gestellt und mich bei Fragen zum *Grundschutzhandbuch* unterstützt hat.

Vielen herzlichen Dank an meine Freundin Katrin. Sie hatte es bestimmt nicht immer leicht mit mir in der Zeit, in der ich an dieser Diplomarbeit geschrieben habe.

Inhaltsverzeichnis

Kapitel 1

Einführung

„Erster Pocket-PC-Virus im Umlauf“ — „Bluetooth-Handys gegen Hacker-Angriffe anfällig“ — „Erster Handy-Wurm entdeckt“ — „Trojaner für Symbian aufgetaucht“

Diese Meldungen sind in letzter Zeit immer häufiger in verschiedenen Newstickern wie z.B. *heise security* zu lesen und lassen viele Fragen über die Sicherheit mobiler Endgeräte aufkommen. „Ist das Endgerät sicher, das ich da benutze?“ „Sind meine Daten geschützt?“ „Was passiert, wenn ich mein mobiles Endgerät verliere?“ Vor allem Unternehmen machen sich darüber verstärkt Gedanken, da die mobile Endgeräte, wie Laptops, PDAs und Mobiltelefone längst in einer Vielzahl in Unternehmen vertreten sind. Auch die Firma *EDAG Engineering & Design AG* in Fulda stellt sich diese Fragen und sucht eine Lösung für diese Probleme.

Diese Diplomarbeit stellt eine Konzeption dar, mit der der Einsatz von mobilen Endgeräten im Unternehmen sicherer gemacht werden kann. Dabei handelt es sich um einen kompletten Lösungsvorschlag für den mobilen Bereich und betrachtet neben der Datensicherheit auch die Datensynchronisation. Denn die Synchronisation mit einer Groupware, wie z.B. *Lotus Notes*, muss ebenso gesichert ablaufen, wie die Speicherung der Daten auf dem jeweiligen mobilen Endgerät. Anforderung der *EDAG Engineering & Design AG* ist es, eine Lösung zu finden, die einerseits für die Benutzer einfach zu bedienen und für die Administratoren einfach zu betreuen ist, und andererseits möglichst alle verschiedenen mobilen Endgeräte und Systeme abdeckt.

Darüber hinaus gibt diese Diplomarbeit Handlungsempfehlungen für die Integration der mobilen Endgeräte in den Standard *BS 7799 / ISO 17799*. Die Handlungsempfehlungen beziehen sich aber auch auf noch bestehende Mängel bei der Umsetzung von *BS 7799 / ISO 17799*. Für diese Mängel müssen bis zum nächsten Audit der Standardisierung Lösungsvorschläge gemacht werden, die durch die Empfehlungen in dieser Diplomarbeit abgedeckt sind.

Kapitel 2

Mobile Endgeräte und Systeme

2.1 Definition mobiler Endgeräte

Mobile Endgeräte sind IT-Systeme, die aufgrund Ihrer Größe unterwegs einsetzbar sind und Aufgaben für eine mobile Kommunikation übernehmen. Diese Endgeräte besitzen eine eigene Stromversorgung, in der Regel eine Batterie, und brauchen keine kabelgebundene Verbindung zu einem Netzwerk. Mobile Endgeräte unterscheiden sich, je nach Größe und Leistung, sehr stark in ihrer Funktionalität. Manche übernehmen komplette Aufgaben eines Desktop-PC's, andere sind nur auf einzelne Aufgaben, wie z.B. Telefonie, beschränkt. Es gibt demnach eine große Mischkultur von verschiedenen mobilen Endgeräten und Systemumgebungen, die nur langsam von der Industrie beseitigt werden. Nach und nach ist man dabei Standard-Umgebungen definiert, die die Verwaltung aber auch die Sicherheit der mobilen Endgeräte verbessern. Davon profitieren nicht nur die Endanwender, sondern vor allem auch die Unternehmen, da der Aufwand diese Endgeräte zu betreuen deutlich geringer wird. [16]

2.1.1 Laptop und Notebook

Ein Laptop[1] ist nichts anderes als ein tragbarer Desktop-PC. Er hat heute die Leistungsfähigkeit gängiger Desktop-Systeme und enthält die gleichen Komponenten, wie z.B. Grafikkarte, Netzwerkkarte, GHz-Prozessor uvm. Die Komponenten sind jedoch speziell für den mobilen Einsatz optimiert, hauptsächlich mit dem Augenmerk, so wenig Strom wie möglich zu verbrauchen. So erreicht man mit aktuellen Endgeräten Laufzeiten von zwei bis sechs Stunden bei einem

[1] engl. laptop=*Auf-dem-Schoß*

Abb. 2.1: *Toshiba Tecra A2*

Stromverbrauch von 13 bis 26 Watt. Notebooks[2] unterscheiden sich von Laptops nur in Größe und Gewicht. So ist ein Notebook in der Regel nicht größer als eine DIN A4 Seite und wiegt zwischen 1kg und 5kg. Laptops können deutlich schwerer sein, da sie größer sind als Notebooks. Sie sind durch sogenannte *PCMCIA*-Karten, auch *PC-Card* genannt, erweiterbar. Durch die *PCMCIA*-Karten können auch ältere Endgeräte von neuen Technologien, wie z.B. *Wireless LAN*, profitieren.

Vor allem in Unternehmen kommen die Laptops immer häufiger zum Einsatz. Ein Grund hierfür ist die flexible Einsatzmöglichkeit der Endgeräte. Man kann überall in der gleichen Weise arbeiten wie mit einem Desktop-PC, kann die gleichen Programme einsetzen und hat somit kaum Einschränkungen in der täglichen Arbeit vor dem Computer. Ein weiterer Grund ist der fallende Preis für diese Endgeräte. Heute sind die Anschaffungs- und Betriebskosten von Desktop-PCs und Laptops nahezu vergleichbar. Lediglich Ersatzteile und Peripheriegeräte sind kostenintensiver, was sich zur Zeit nach und nach an die Preise für die Erweiterungen für Desktops-PCs angleicht. Ein letzter hier erwähnter Grund für einen verstärkten Einsatz von Laptops ist die technische Entwicklung. In keinem Bereich der IT wird derzeit mehr Aufwand für die Weiterentwicklung betrieben, als bei den Laptops. Dies liegt nicht nur an der Entstehung von neuen Technologien wie Intels *Centrino* oder neuen Netzwerk-Standards wie *IEEE 802.11i*[3].

U.a. durch diese technischen Neuerungen entstehen aber auch neue Gefahren für die Sicherheit der mobilen Endgeräte und daran angeschlossenen Netzwerken. Somit ist der Einsatz von Laptops und Notebooks in Unternehmens-Netzwerken stets mit Vorsicht zu betrachten und gesonderte Regeln beim Einsatz dieser Endgeräte zu definieren.

2.1.2 PDA

Anfang der 1990er Jahre wurden einige Studien über den Einsatz von Computern in der Zukunft erstellt. Dabei war eine Studie auch die Idee, einen kleinen Computer, den *Knowledge Navigator*, zu entwickeln, der alle Aufgaben einer Sekretärin, wie z.B. Anrufe entgegen nehmen, Termine verwalten und Nachrichten und Notizen schreiben, übernehmen soll. Aus der Studie des *Knowledge Navigators* machten die Entwickler von Apple das *Newton MessagePad*, welches die neue Kategorie der PDA begründete.

Abb. 2.2: *Apple Newton MessagePad 2000*

Das *Newton MessagePad* war im Vergleich zu heutigen PDAs ein recht großes und unhandliches Endgerät. Es gab z.B. einen Prototyp des *Newton MessagePads* der mit einem CD-Laufwerk ausgestattet war. Das *MessagePad* hatte einen 20 MHz ARM-Prozessor und bis zu 4 MB ROM- und 640 kb RAM-Speicher und war mit einer PCMCIA-Schnittstelle ausgestattet. Das monochrome *Touch-LCD-Display* hatte eine Auflösung von 336 × 240 Pixeln, die Eingabe erfolgte über einen spezi-

[2] engl. notebook=*Notizbuch*

[3] aktueller WirelessLAN-Standard

ellen Stift direkt auf das Display.

Hauptaustattung im *Newton MessagePad* war damals schon die PIM-Software mit der Kontaktdaten, Termine, Aufgaben und Notizen erfasst werden konnten. Natürlich war die Funktionalität über eigene Progamme erweiterbar. Darüber hinaus verfügte der *Newton* über eine multitaskingfähige *Graphical User Interface* (*GUI*, die in einer Fensterstruktur alle laufenden Prozesse anzeigte. Das revolutionäre am *Newton MessagePad* war jedoch die Entwicklung einer Handschrifterkennung. Diese war aber, vor allem bei den ersten *Newtons*, noch sehr ungenau und führte sehr oft zu einer falschen Erkennung. Erst in nachfolgenden Modellen wurde die Schrifterkennung immer besser. Dieses Problem und der hohe Preis von durchschnittlich 700 bis 1.000 US-$ mag vielleicht der Grund dafür gewesen sein, dass nur ca. 300.000 *Newtons* verkauft und die Produktion 1998 eingestellt wurde. Dennoch sind die *Newtons* heute im täglichen Einsatz und haben eine große Fan-Gemeinde. [2]

Durch die Produktionseinstellung der *MessagePads* wechselnten viele *Newton*-Entwickler zur *Palm Computing Inc.*, damals ein Tochterunternehmen von *U.S. Robotics*, und machten dort die Idee des PDAs massentauglich. *Palm Computing Inc.* brachte damals den *Pilot*, später *PalmPilot*, auf den Markt, der mit der neuen Schrifterkennungs-Software *Graffiti* ausgestattet war. Ein Merkmal von *Graffiti* sind standardisierte Zeichen, die die Genauigkeit bei der Schrifterkennung merklich erhöhen. Somit muss sich der Mensch auf den PDA einstellen und nicht umgekehrt, wie es beim *Newton MessagePad* der Fall war. Noch heute wird man bei der Einrichtung eines aktuellen Palm-Endgerätes zuerst mit der *Graffiti*-Schrifterkennung vertraut gemacht, bevor man das Endgerät überhaupt benutzen kann.

Spätestens nach der Übernahme von *U.S. Robotics*, und dadurch auch von *Palm Inc.*, durch *3Com* starteten die *PalmPilot*s ihren Siegeszug. Der *3Com Palm III* war der Verkaufsschlager Ende der 1990er Jahre und kam in unzähligen Varianten auf den Markt. 1999 war *3Com* mit den *PalmPilot*s bereits Weltmarktführer bei den PDAs. Zu dieser Zeit war *Palm* das Synonym für PDAs und das *Palm OS* Quasi-Standard für alle Endgeräte dieser Art.

Abb. 2.3: *3Com Palm III*

Ebenfalls Ende der 1990er Jahre kam auch *Microsoft* auf die Idee ein System für mobile Endgeräte zu entwickeln und veröffentlichte sein Betriebssystem *Windows CE*. *Microsoft* schaffte es erst im Jahr 2000 mit dem *Handheld PC*, später *PocketPC*, auf dem PDA-Markt konkurrenzfähig zu werden. Dabei stammte jeweils nur das Betriebssystem von *Microsoft* und die Hardware wurde von Lizenznehmern, wie z.B. *Hewlett-Packard*, *Toshiba* oder *Fujitsu-Siemens*, entwickelt. Es gibt bis heute kein Endgerät, das komplett von *Microsoft* entwickelt und auf den Markt gebracht wurde.

Aktuelle PDAs stehen Laptops heute nur noch in wenigen Punkten nach. Bluetooth und WirelessLAN finden in PDAs ebenso Verwendung wie mobile Speicherkarten, z.B. SD-Card, MemoryStick oder CompactFlash. Die Endgeräte sind über die CompactFlash- oder SDIO-Schnittstelle jederzeit durch neue Technologien erweiterbar. Ihre Hauptaufgabe hat sich aber

seit dem *Newton MessagePad* nicht verändert — sie werden weiterhin hauptsächlich für die Speicherung von Kontaktdaten, Terminen und Nachrichten verwendet.

2.1.3 Mobiltelefon und Smartphone

Seit Mitte der 1990er Jahre drängt verstärkt eine weitere Kategorie der mobilen Endgeräte auf den Markt — die Mobiltelefone. Umgangssprachlich sind die Mobiltelefone eher als *Handy* bekannt. Bis Mitte der 1980er Jahre waren die Mobiltelefone fest in Fahrzeugen eingebaut und wurden somit als Autotelefon verwendet. Unter anderem durch die Einführung des analogen C-Netzes (Dezember 1988) wurden die Mobiltelefone immer unabhängiger von ihrem Einsatzort. Damals waren die aktenkoffergroßen Telefone noch sehr unhandlich und meist unerschwinglich. Das erste kommerzielle Mobiltelefon, das *Motorola DynaTAC 8000X*, kostete 1983 ca. 4.000 US-$ und wurde wegen seiner Größe und seines Aussehens von allen nur liebevoll als *Knochen* bezeichnet.

Abb. 2.4: *Motorola DynaTAC 8000X*

Mit der Einführung des digitalen D-Netzes und dem *Global System for Mobile Communication* (*GSM*) Anfang der 1990er Jahre wandelte sich aber der Einsatz der Endgeräte. Der finnische Hersteller *Nokia* war einer der Vorreiter in der flächendeckenden Einführung der neuen Technologie. Die Mobiltelefone wurden kleiner, leistungsfähiger und darüber hinaus billiger und verbreiteten sich zum Jahrtausendwechsel rasant. Heute sind ca. 1,2 Milliarden Mobiltelefone weltweit im Gebrauch und aus dem täglichen Leben nicht mehr weg zu denken. Konnte man damals mit den Mobiltelefonen lediglich telefonieren und seit 1993 auch Textnachrichten (*SMS*) versenden. Ausgestattet mit Digitalkameras oder der Unterstützung von Spielen und Multimedia-Anwendungen sind Mobiltelefone heute wahre Unterhaltungsmaschinen. Inwiefern diese Funktionalitäten sinnvoll sind oder hier eine Zweckentfremdung statt findet, ist jedem selbst überlassen. Darüber hinaus werden aber auch nützliche Anwendungen, wie z.B. die Verwaltung von Kontaktdaten, Terminen und Nachrichten, enthalten und das Mobiltelefon bietet dadurch ähnliche Funktionalitäten, wie bei einem PDA. Technologien wie Bluetooth und WirelessLAN halten ebenfalls immer mehr Einzug in die kleinen Endgeräte.

NOKIA
CONNECTING PEOPLE

Abb. 2.5: *Logo Nokia*

Ein Problem an den Mobiltelefonen ist aber, dass in der Regel jeder Hersteller sein eigenes, genau auf die Hardware zugeschnittenes Betriebssystem verwendet. Ein Vorteil davon ist, dass Mobiltelefone dadurch relativ sicher gegenüber Angriffen durch fehlerhaften Code o.ä. sind. Dadurch ergeben sich allerdings auch Probleme z.B. in der Datensynchronisation oder bei der Administrierung. Viele Unternehmen sind deswegen dazu über gegangen nur Endgeräte von einem Hersteller anzuschaffen und andere Mobiltelefone nicht mehr zuzulassen. Die Hersteller haben sich in der letzten Zeit darauf geeinigt, Funktionen in ihre

Betriebssysteme zu integrieren, die wesentliche Aufgaben wie Datenzugriff und Datensynchronisation standardisieren. Für die Datensynchronisation haben sich mittlerweile *SyncML* und *Object Exchange* (*OBEX*) etabliert und werden von nahezu allen Herstellern in deren Betriebssystemen unterstützt. *OBEX* ist beispielsweise von der *IrDA* für den allgemeinen Datenaustausch von PIM-Daten bis hin zum normalen Datei-Transfer standardisiert. *SyncML* setzt rein auf die Synchronisation der PIM-Daten und wird von der *Open Mobile Alliance*, bestehend aus vielen Hersteller, wie z.B. *Nokia*, *Sony Ericsson*, *Palm*, *Motorola*, *IBM* uvm., standardisiert[4].

1996 entwickelte *Nokia* das erste *Smartphone* — die Verschmelzung von PDA und Mobiltelefon — und brachte den *Nokia Communicator 9000* auf den Markt. Im *Nokia Communicator* kommt dabei *Symbian OS* zum Einsatz, das von *Nokia*, *Sony Ericsson* und weiteren Mobilfunkherstellern entwickelt wird und bislang rein auf den Einsatz in Smartphones spezialisiert ist. Aber auch *Palm OS* findet seine Verwendung in Smartphones. Vor allem die *Handspring Treo*s setzen auf dieses System. Microsoft hat ebenfalls spezielle Versionen von *Windows CE* entwickelt und bietet das *Windows Smartphone 2002* und *Windows Mobile PhoneEdition* an. Dabei gewinnen die Windows-Endgeräte, unter anderem wegen der guten Zusammenarbeit zwischen den mobilen Endgeräten und den Windows-Desktops, immer mehr an Beliebtheit bei den Endanwendern.

Abb. 2.6: *Nokia 9110 Communicator*

Marktführer sind derzeit jedoch die *Symbian OS*-basierten Endgeräte. Seit Anfang 2004 drängt noch eine weitere Kategorie der Smartphones auf den deutschen Markt, der *BlackBerry*. Diese Endgeräte bestechen durch ihre mobilen Möglichkeiten bei der Datensynchronisation. Bei der *EDAG Engineering & Design AG* ist der Einsatz der *BlackBerry*-Technologie jedoch nicht geplant und wird in dieser Diplomarbeit nur der Vollstänsigkeit halber erwähnt.

2.2 Mobile Betriebssysteme

Die große Anzahl an unterschiedlichen mobilen Endgeräten hat natürlich auch zur Folge, dass viele verschiedene Betriebssysteme existieren, die in den Endgeräten zum Einsatz kommen. Für die Mobiltelefone gibt es pro Hersteller, oft auch je Mobiltelefon ein eigenes Betriebssystem, das genau auf die Hardware zugeschnitten ist. Auch für Smartphones und PDAs existiert kein einheitliches Betriebssystem, das generell zum Einsatz kommt. Bei den Laptops ist die Auswahl schon kleiner — aber auf keinem Fall einheitlich — und von den Unternehmen abhängig, in denen die Laptops zum Einsatz kommen. Jedes einzelne dieser Systeme hat ebenfalls seine Eigenheiten und Schwachstellen, die Einfluss auf die Sicherheit haben können.

[4]Nähere Infos zu SyncML:
siehe http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html

2.2.1 Universalbetriebssysteme

2.2.1.1 Microsoft Windows

Microsoft Windows ist bereits seit Ende der 1980er Jahre auf Desktop-PCs im Einsatz und hat sich dort vor allem im Endanwender-Bereich durchgesetzt. Auch bei Unternehmen ist *Microsoft Windows* ein Betriebssystem, das sich zu einem Quasi-Standard entwickelt hat. Derzeit ist *Microsoft Windows* auf ca. 80% aller eingesetzten Desktop-PCs installiert. *Windows* ist dabei die Weiterentwicklung von *MS-DOS*, die auf das vorhandene Betriebssystem anfänglich nur eine *GUI* aufsetzte. *MS-DOS* war noch bis zu *Windows ME* das Grundgerüst des Betriebssystems. Erst mit *Windows XP* verabschiedete sich *Microsoft* endgültig von der *MS-DOS*-Architektur. Heute befinden sich hauptsächlich *Windows 2000* und *Windows XP* bei Unternehmen im Einsatz. Beide Betriebssysteme basieren auf der NT-Technologie. *NT* bietet eine größere Stabilität und eignet sich deswegen sehr gut für den Einsatz in Unternehmen. Alle Server-Versionen von *Microsoft* basieren auf dieser Technologie. Im Folgenden soll nur *Microsoft Windows XP* etwas näher betrachtet werden, da dieses Betriebssystem sich nicht wesentlich von *Windows 2000* unterscheidet und beide Betriebssysteme heute am häufigsten im Einsatz sind. Der Einsatz von *Windows XP* wird von *Microsoft* empfohlen und ist auf den meisten Computern bereits vorinstalliert. *Windows XP* ist bei den Laptops meist als Standardsystem installiert, da dieses Betriebssystem zur Zeit die beste Unterstützung für tragbare Computer zur Verfügung stellt.

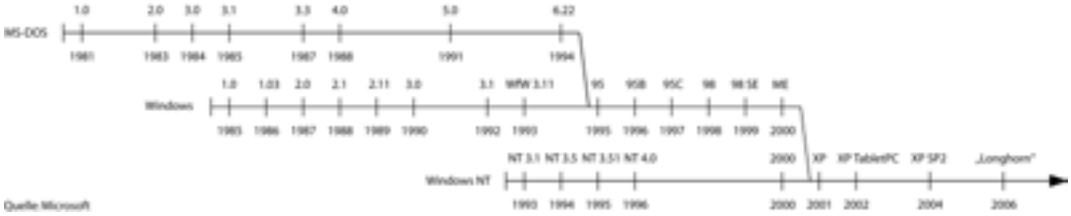


Abb. 2.7: *Entstehungsgeschichte von Microsoft Windows*

Windows XP ist ein multitasking- und multithreading-fähiges Betriebssystem und existiert in den Versionen *Windows XP Home* und *Windows XP Professional*. Die *Home*-Version ist vornehmlich für private Endanwender gedacht, die zu Hause einen einzelnen Rechner oder einen kleinen Rechnerverbund haben. Die Netzwerkfunktionen von *Windows XP Home* sind sehr eingeschränkt und somit in Unternehmensnetzwerken kaum einsetzbar. Die *Professional*-Version ist für Unternehmen deswegen besser geeignet. *Windows XP* ist nach einem Schichtenmodell in — vereinfacht gesehen — vier Stufen aufgebaut. Auf der untersten Stufe steht der *Hardware Abstraction Layer* (*HAL*). Der *HAL* stellt dabei die nötigen Treiber für die verwendete Hardware zur Verfügung. *Windows XP* unterstützt ausschließlich Prozessoren mit x86-Architektur, sog. Intel-kompatible Prozessoren. Darunter gehören neben den Prozessoren von *Intel* auch die Prozessoren von *AMD*. Ansonsten unterstützt *Windows XP* problemlos jede PC-Hardware die derzeit auf dem Markt ist. Die Hersteller von PC-Hardware setzen wegen der großen Verbeitung von *Microsoft Windows* teilweise aus-

schließlich auf die Unterstützung dieses Betriebssystems. Man kann davon ausgehen, dass zu jeder PC-Hardware auch ein entsprechender Windows-Treiber verfügbar ist. Darüber hinaus enthält *Windows XP* eine große Datenbank von Standard-Treibern, mit denen jegliche Arten von PC-Hardware, wie z.B. Netzwerkkarten, Grafikkarten, Eingabegeräte und Drucker, zumindest in einfacher Weise, angesprochen werden können. Auf der zweiten Stufe befindet sich die *Betriebssystems-Schicht* mit dem Kernel und allen wichtigen Systemdiensten wie das Speichermanagement, dem Fenstermanagement und dem Eingabe/Ausgabe-Management. Auf der *Betriebssystem-Schicht* setzt nun das *Win32-Subsystem* auf, dass eine einheitliche Schnittstelle für die Entwickler von Windows-Applikationen bietet. Für die Entwicklung von Applikationen stellt Microsoft die *Win32 API* zur Verfügung, die standardisierte Funktionen enthält, um auf das darunter liegende Betriebssystem und durch Treiber bereit gestellte Hardware zuzugreifen. Die oberste Stufe ist die *Applikations-Schicht* in der die eigentlichen Anwendungen laufen und mit der der Endbenutzer täglich zu tun hat.

Abb. 2.8: *Aufbau von Microsoft Windows XP*

Windows XP steht leider sehr oft wegen Sicherheitslücken und Sicherheitsproblemen in den Schlagzeilen. In *Windows XP* sind aber einige Sicherheitsfunktionen eingebaut, die eine grundlegende Sicherheit für ein Windows-System zur Verfügung stellen. *Windows XP* bietet einmal die klassischen Sicherheitsfunktionen für die Datenübertragung wie *SSL* oder *TLS*. In *Windows XP* ist auch erstmals eine Firewall direkt eingebaut, mit der man einfache grundlegende Regeln definieren kann, um den Computer vor unerlaubten Zugriffen zu schützen. Als Dateisystem kann man zwischen *FAT* und *NTFS5* wählen. Teil des *NTFS5* ist das *Encrypted File System* (*EFS*). Das *EFS* bietet eine einfache Möglichkeit Daten und Datenträger zu verschlüsseln. Ausgenommen hiervon sind Systemdateien. Für jede zu verschlüsselnde Datei wird ein Schlüssel von 128bit Länge erzeugt, der mit dem öffentlichen Schlüssel des Benutzers und einem öffentlichen Schlüssel eines „Wiederherstellungs-Agenten“ verschlüsselt wird, der im Notfall die verschlüsselte Datei wieder herstellen kann [26, 27]. Ein Problem an *EFS* ist, dass die Daten unverschlüsselt im Hauptspeicher bzw. in der temporären Auslagerungsdatei liegen, wenn sie z.B. in eine Applikation geladen werden. Nur solange die Dateien nicht genutzt werden, sind sie auf dem Computer sicher. Es gibt mittlerweile viele Beschreibungen für *Brute-Force*-Angriffe auf das EFS, was einen Einsatz unsicher macht.

Außerdem gibt es eine Vielzahl an Viren, Würmern und Trojanern, die zusammenfas-

send auch als *MalWare*[5] bezeichnet werden, die explizit Schwachstellen in den *Windows*-Betriebssystemen ausnutzen. Ein Grund dafür ist nicht nur ein evt. schlecht programmiertes Betriebssystem, sondern oftmals eher schlecht programmierte Applikationen. Diese Applikationen öffnen Schwachstellen, wie z.B. Ports oder andere Benutzerrechte, die von der *MalWare* ausgenutzt werden. Bekannte Beispiele für *MalWare* die solche Schwachstellen ausnutzen sind *W32.Sasser*, *W32.MyDoom*, *W32.NetSky* oder *CodeRed*. *Microsoft* versucht diese Schwachstellen durch monatliche Sicherheitspatches oder sog. *ServicePacks* zu beseitigen. Momentan gibt es für *Windows XP* das *ServicePack 2*, das *Windows XP* mit besseren Sicherheitsfunktionen ausstattet. So wurde die eingebaute Firewall um viele Funktionen erweitert und muss sich hinter anderen Personal Firewalls nicht mehr verstecken. Das neue *Sicherheitscenter* überwacht neben der Windows-Firewall sogar ein installiertes Antiviren-Tool auf Aktualität und warnt den Benutzer wenn der Virenschutz nicht mehr gewährleistet wird. *Microsoft* reagiert somit verstärkt auf die Gefahren aus dem Internet und bietet mit seinem Betriebssystem einen gewissen Grundschutz vor *MalWare*.

Microsoft entdeckte 1996 auch den Markt für die kleinen Computer und begann mit der Entwicklung seines Betriebssystems *Windows CE*[6]. *Windows CE* hat eine schlanke Architektur und findet, im Gegensatz zu seinem oben beschriebenen großen Bruder, auf Endgeräten mit eingeschränkten Ressourcen, wie Festplattenspeicher und RAM, Platz. Dabei stellt *Windows CE* nur grundlegende Betriebssystem-Funktionen wie Thread-Verwaltung, Zugriff auf die Ressourcen usw. zur Verfügung. Alles weitere, wie z.B. eine GUI fehlen zunächst komplett und müssen von den Entwicklern für ihr Endgerät selbst erstellt werden. Microsoft bietet dazu den *Microsoft Platform-Builder* an, mit dem man definiert, welche Funktionalitäten, z.B. Bluetooth- oder WirelessLAN-Unterstützung, das Betriebssystem für das entwickelte Endgerät zur Verfügung stellen soll. Für die Entwicklung von Anwendungen für *Windows CE* stellt Microsoft die *eMbedded Visual Tools* und spezielle *Software Development Kits* (*SDK*) kostenlos zur Verfügung.

Windows CE ist die Grundlage vieler Kleincomputer und kommt z.B. in Videorecordern, Registrierkassen, Industriemaschinen und eben in PDAs und Smartphones zum Einsatz. *Windows CE* wird bereits seit der Version 1.0 in PDAs in der sog. *Handheld PC Edition* eingesetzt. Seit der Version 3.0 wird das Betriebssystem nur noch *Handheld PC*, später *PocketPC*, genannt. Ab Version 4.2 bezeichnet *Microsoft* es als *Windows Mobile 2003*. Mit der kommenden Version 5.0 geht *Microsoft* aber wieder zu der Bezeichnung *Windows CE 5.0* zurück, um den bestehenden Verwirrungen bei den Endanwendern entgegen zu wirken.

Der Aufbau von *Windows CE* erfolgt, wie schon bei *Windows XP*, in einem Schichtenmodell. Dabei gibt es — von unten nach oben — die Schichten *Hardware*, *OEM*, *Betriebssystem* und *Applikation*. In der *Hardware-Schicht* befinden sich alle Schnittstellen wie Bildschirm, Netzwerk-Schnittstellen, Eingabegeräte — Maus, Tastatur oder Stift — oder Speichermedi-

[5]Die Bezeichnung *MalWare* setzt sich zusammen aus *malicious* und *Software* und bedeutet so viel wie *fehlerhafte Software*

[6]Das *CE* ist laut Microsoft keine Abkürzung, sondern einfach nur der Produktname. *siehe* http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q166915

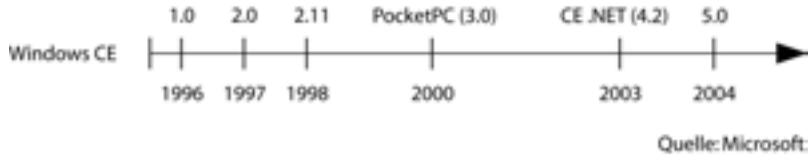


Abb. 2.9: *Entstehungsgeschichte von Windows CE*

en. *Windows CE* unterstützt alle gängigen Mikroprozessoren, wie *ARM*, *MIPS*, *SHx* und *x86*. Als Netzwerkschnittstellen können Firewire, Bluetooth, IrDA oder WirelessLAN zum Einsatz kommen. Die *OEM-Schicht* dient dazu, den Zugriff auf die verfügbare Hardware durch das Betriebssystem zu ermöglichen. Dafür können die Hersteller eigene Treiber für *Windows CE* entwickeln. Daneben steht ein *OEM Adaption Layer* über den spezielle Boot-Loader und Konfigurationsdateien definiert werden. Das eigentliche Betriebssystem mit Kernel, Event-System, Endgerätemanager und Multimedia-Unterstützung befindet sich in der *Betriebssystem-Schicht*. Der Kernel ist, wie bei allen Windows-Versionen, multitasking- und multithreadingfähig, was ein paralleles Ausführen von Applikationen ermöglicht. Daten können bei *Windows CE* entweder ROM- oder RAM-basiert oder auf externen Speichermedien, wie beispielsweise *CompactFlash*-Karten, *FAT*-basiert gespeichert werden. Über der *Betriebssystem-Schicht* steht die *Applikations-Schicht*. Hier werden alle Applikationen abgelegt, die für das entwickelte Endgerät notwendig sind. Für einen PDA wären das z.B. die PIM-Anwendungen, bei einem Smartphone zusätzlich die Telephonie-Unterstützung. [17]

Die Datensynchronisation findet bei *Windows CE* über das Programm *ActiveSync* statt. Dazu muss der *Windows CE*-PDA an einen Host-Rechner, vornehmlich einem Windows-Rechner, angeschlossen sein, auf dem das Programm installiert ist. *ActiveSync* kann sowohl mit einer einfachen eMail- und Kontaktverwaltungs-Anwendung, wie *Microsoft Outlook Express* bzw. mit einer professionelle Groupware-Anwendung, wie *Microsoft Outlook*, als auch mit Groupware-Servern, wie *Microsoft Exchange Server* oder *Lotus Notes Server*, synchronisieren. Der Benutzer muss bei der Installation nur Regeln darüber definieren, was in welchem Umfang synchronisiert werden soll. Sind diese Einstellungen einmal gemacht, führt *ActiveSync* die Synchronisation als Standardeinstellung vollkommen automatisch durch.

Abb. 2.10: *Screenshot von ActiveSync*

Ein Highlight an *Windows CE* ist die Handschrifterkennung *Transcriber*. Hier nimmt Microsoft wieder das Konzept des *Newton MessagePads* auf und interpretiert die normale Schrifteingabe des Benutzers. Der Benutzer muss also nicht erst eine gesonderte Schriftsprache erlenen, bevor er Einträge in das Endgerät machen kann, sondern kann sofort damit anfangen. Eine deutliche und saubere Handschrift ist dabei aber Voraussetzung, sonst könnte *Transcriber* Probleme haben, das Geschriebene zu erkennen. Die Schrifterkennung geschieht aber wesentlich besser als bei den *Newton MessagePads*. *Transcriber* hat zusätzlich die Möglichkeit, sich nach und nach auf die Schreibweise des Benutzers einzustellen. Dazu verwendet *Transcriber* eine hochentwickelte Fuzzy-

Logik, die Eingaben in Schreibschrift, Druckschrift oder einer Kombination ermöglicht.

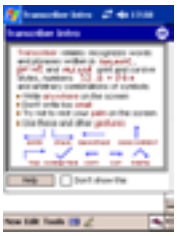


Abb. 2.11: *Transcriber Schriftbilderkennung*

Sicherheitsfunktionen sind nur sehr eingeschränkt in *Windows CE* vorhanden. Neben der möglichen Verschlüsselung der Datenübertragung durch *SSL* bzw. *TLS* und der Authentifizierungsverfahren *Kerberos* und *NTLM* bietet *Windows CE* lediglich die Möglichkeit, den PDA mit einem selbst definierten alphanumerischen Passwort zu schützen. Eine direkte Verschlüsselung der Daten auf dem PDA ist nicht vorgesehen. Auch in der neuen Version *Windows CE 5.0* ist kein Ansatz für eine solche Unterstützung zu finden.

Im PDA-Bereich gewinnt *Windows CE* langsam an Marktanteilen und hat im dritten Quartal 2004 erstmals *Palm OS* überholt. Demnach haben die *Windows CE*-basierten Endgeräte einen Marktanteil von ca. 48% und die *Palm OS*-basierten Endgeräte liegen auf Platz 2 mit ca. 30% [12]. Das hängt insbesondere mit der guten Zusammenarbeit zwischen den Desktop-Versionen von Windows und der vertrauten Bedienung, die schon von den Desktop-Versionen bekannt ist, zusammen.

2.2.1.2 Linux

Ende 1991 veröffentlichte Linus Torvalds das Unix-ähnliche und POSIX[7]-kompatible Betriebssystem *Linux*. Mit *Linux* ist eigentlich nur der Kernel gemeint, die Bezeichnung hat sich heute aber für das komplette Betriebssystem durchgesetzt. *Linux* steht unter der *General Public License* (*GPL*) und der Quellcode ist somit jedem frei zugänglich. Deswegen arbeiten eine Vielzahl von Entwicklern an neuen brauchbaren Funktionen und verbessern unter der Leitung von Linus Torvalds weiter den Kernel. Aktuell ist Version 2.6 des *Linux*-Kernels.

Linux wird von Unternehmen hauptsächlich für den Serverbereich eingesetzt. Dort ist *Linux* als stabiles und sicheres Betriebssystem bekannt. Der Einsatz als Desktop-Version ist vor allem durch die Linux-Distributionen vorangetrieben worden. Eine Linux-Distribution besteht neben dem Linux-Kernel aus einer GUI, bei *Linux X-Window-System* genannt, wie z.B. *KDE* oder *Gnome*, sowie einer Menge nützlicher Applikationen, die oftmals ebenfalls unter der GPL stehen. Die Applikationen reichen von kompletten Office-Suiten (*OpenOffice*) bis hin zu professionellen Grafikprogrammen (*Gimp*). Eine Linux-Distribution ist nicht selten auf mehreren CDs oder DVDs untergebracht. Die wichtigsten Linux-Distributionen sind u.a. *SuSE Linux* für den deutschsprachigen Raum und *RedHat Li-*

Abb. 2.12: *Logos der größten Distributoren*

[7] **P**ortable **O**perating **S**ystem **I**nterface for Uni**x**

nux für den amerikanisch-englischen Raum. Diese kommerziellen Distributionen sind wegen der Zusammenstellung von Handbüchern, Installationsmedien und der Bereitstellung von Support zwar nicht kostenlos, aber in der Anschaffung weitaus preiswerter als ein *Microsoft Windows* und haben schon allein durch die Vielzahl von beigelegten Applikationen einen gewissen Mehrwert. Da die Distributionen in der Regel frei kopierbar sind (*GPL*), reicht meist die Anschaffung von nur einer Distribution, die dann im ganzen Unternehmen eingesetzt werden kann. Im Zweifelsfall sollte aber gepfrüft werden, in wiefern der jeweilige Distributor ein eigenes Lizenzmodell anbietet. Die am weitesten verbreitete freie Distribution ist *Debian GNU/Linux*, kurz *Debian*, die auf Basis der *GPL* zur Verfügung gestellt wird. Durch die große Anzahl an unterschiedlichen Linux-Distributionen gibt es aber auch einige Probleme. So müssen alle Applikationspakete zunächst für die jeweilige Distribution kompiliert werden, damit die Applikation auf dieser Distribution lauffähig ist. Als *Linux* auf den Markt kam war dieser Prozess Gang und gebe. Heute übernehmen Installationsmanager diese Aufgabe, insofern bereits vorkompilierte Applikationspakete für die Linux-Distribution existieren. Dabei gibt es aber keinen einheitlichen Installationsmanager, sondern jede Distribution hat ihren eigenen. Bei *RedHat Linux* gibt es z.B. den *RPM Package Manager* (*RPM*), der auch von *YaST* von *SuSE Linux* zur Installation verwendet wird. Bei *Debian* kommt das *Advanced Packaging Tool* (*APT*) für den Installationsprozess zum Einsatz. Vom *APT* steht auch eine Version für *RPM*-Pakete zur Verfügung. Auch an anderen Stellen gibt es einige Inkompatibilitäten zwischen den Linux-Distibutionen, so dass sich die Linux-Gemeinschaft dazu entschlossen hat mit dem *Linux Standard Base* eine weitere Versplitterung zu vermeiden. Viele der Linux-Distributoren haben versprochen, sich an diesen Standard zu halten und dessen Entwicklung mit voran zu treiben. Der *Linux Standard Base* definiert beispielsweise eine einheitliche Dateistruktur oder grundsätzlich notwendige Bibliotheken. Das nächste Ziel ist die Verbesserung der Zusammenarbeit zwischen *RPM*- und *Debian*-basierten Distributionen.

Durch die Standardisierungen vermindert sich der administrative Aufwand für einen Einsatz von Linux auf Desktops, was früher nur professionellen Anwendern vorbehalten war. Somit rückt Linux allein schon wegen dem Lizenzmodell und den geringeren Kosten immer mehr in das Interesse von Unternehmen. Viele Behörden, wie die Münchner Stadtverwaltung oder der Deutsche Bundestag, haben bereits angefangen Linux auf den Desktops ihrer Beschäftigten als Standard-System einzusetzen. Dass die Verbreitung von *Linux* allerdings noch nicht weiter fortgeschritten ist, liegt wohl auch an der mangelnden Verfügbarkeit vieler Anwendungen. So vertrauen nur wenige Firmen unternehmenskritische Anwendungen wie *SAP* dem freien Betriebssystem an.

Abb. 2.13: *Sharp Zaurus*

Auch bei mobilen Endgeräten sucht man *Linux* als Betriebssystem vergeblich. Mit der speziellen Hardware von Laptops kommen viele Linux-Distributionen nicht zurecht und verweigern dort regelrecht ihren Dienst. Auch in PDAs und Smartphones ist *Linux* eher ein seltener Kandidat. Zwar gibt es Versuche von PDA-Herstellern Endgeräte mit *Linux*, z.B. der *Sharp Zaurus*, auf den Markt zu etablieren, was aber bisher mißlang. Seit Anfang 2003 entwickelt eine Community die *Familiar Distribution*[8] und das damit verbundene *Open Palmtop Integrated Enviroment* (*OPIE*), ein auf *Debian GNU/Linux*-basiertes

[8] *siehe* http://familiar.handhelds.org

Betriebssystem für PDAs und Smartphones. Derzeit existieren Versionen für den *Hewlett Packard iPAQ* und den *Sharp Zaurus*. Auch Anleitungen für die Installation auf anderen mobilen Endgeräten sind dort hinterlegt. Somit finden auch Hersteller wieder zurück in die Verwendung von *Linux* als Betriebssystem und so hat z.B. *Motorola* schon ein *Linux*-Smartphone auf den Markt gebracht und bereits weitere angekündigt. Dennoch bleibt Linux eher Außenseiter beim Einsatz in mobilen Endgeräten und wird daher in dieser Diplomarbeit nur am Rande betrachtet. Die Administrierbarkeit mag aber bei aktuellen *Linux*-Distributionen bestimmt weniger an Aufwand bedeuten als bei manch anderen Systemen. Grund hierfür sind die bessere Unterstützung von neuen Technologien, wie z.B. *WLAN* oder *Bluetooth*, und aktueller Hardware. Bei älteren Distributionen war dies nur durch einen sehr hohen administrativen Aufwand möglich.

2.2.2 Systeme für PDA und Smartphones

2.2.2.1 Palm OS

Insbesondere weil *Palm OS* das erste marktfähige Betriebssystem für mobile Endgeräte war, kommt es heute in ca. 2/3 der PDAs zum Einsatz und wird von 85% der Unternehmen als Standard-Betriebssystem für PDAs definiert. Seit 1996 wird *Palm OS* stetig weiter entwickelt und die Version 6 steht bereits in den Startlöchern — das *Palm OS Cobalt*.

Palm OS bietet eine Handschrifterkennung über die standardisierte Eingabesprache *Graffiti*. Dabei müssen vorgegebene Strichfolgen in einen speziellen Bereich des Display eingegeben werden. Die *Graffiti*-Sprache ist zwar gewöhnungsbedürftig, aber sehr leicht zu erlernen. Alternativ gibt es auch eine Software-Tastatur über die die Eingaben gemacht werden können. Einige Hersteller bieten für ihre Endgeräte auch externe Hardware-Tastaturen an, die den Eingabe-Komfort natürlich noch mehr erhöhen.

Als Schnittstelle zu anderen Rechnern, wie z.B. Laptops oder Desktop-PCs, steht den Palm-Endgeräten ein serieller oder USB-Anschluss zur Verfügung. Darüber hinaus haben die Palm-Endgeräte auch eine Infrarotschnittstelle nach dem IrDA-Standard. Erst seit *Palm OS Version 5* verfügen die Palm-Endgeräte über die Unterstützung von Bluetooth oder Wireless LAN.

Abb. 2.14: *Schriftbild von Graffiti*

Ist das Palm-Endgerät mit einem Laptop oder Desktop-PC verbunden, dann geschieht die Datensynchronisation über die *HotSync*-Software. Über *HotSync* kann auch zusätzliche Software installiert und die kompletten Daten des Palms gesichert werden. *HotSync* ist für *Microsoft Windows*, *MacOS X* und *Linux* verfügbar. Ist der Palm mit dem Host-Rechner verbunden, findet die Synchronisation automatisch statt. Andernfalls muss

man sie über die *HotSync*-Software im *Palm OS*-Endgerät manuell starten.

Abb. 2.15: *Logo HotSync*

Im Bereich der Sicherheit bietet das *Palm OS* bis zur Version 5 nur einen einfachen Passwortschutz. Dabei wird im Endgerät ein alphanumerisches Passwort hinterlegt, das beim Einschalten des Endgerätes eingeben werden muss. Darüber hinaus kann man festlegen, nach welcher Zeit sich das Endgerät selbst sperrt. Eine Verschlüsselung der Daten ist nicht implementiert.

Erst für *Palm OS Cobalt* sind weitere Sicherheitsfunktionen geplant. Kernstück wird der *Cryptographic Provider Manager (CPM)* sein, der ein *Application Programming Interface* (*API*) für verschiedene kryptografische Operationen zur Verfügung stellt. Die API bietet Funktionen für die Generierung von Schlüsseln und Hash-Werten, Verschlüsselung, Entschlüsselung, Signierung und Verifikation. Als kryptografische Verfahren kommen *RC4*indexKryptografie!Verfahren!RC4 (Schlüssellänge 128bit), *SHA-1*[9] und *RSA*[10] (Schlüssellänge 1024bit) zum Einsatz. Ein weiteres Modul in der Sicherheitslösung von *PalmOS Cobalt* ist der *Certificate Manager*, der die Verwaltung von *X.509* Zertifikaten übernimmt. Für eine sichere Verbindung ist zusätzlich *SSL* v2, v3 und *TLS* 1.0 implementiert. Als letzte Komponente kommt das *Security Services* Modul ins Spiel. Diese Modul ermöglicht es für das Palm-Endgerät *Policies*, also Regeln, zu erstellen, die die Funktionsweise des kompletten Endgeräts bzw. auch nur einzelner Applikationen genau definiert. Die *Policies* bestimmen dann, ob der Benutzer z.B. ein WirelessLAN-Modul benutzen darf oder nicht. Dabei wird beispielsweise definiert, welche Applikationen und/oder Treiber auf dem Endgerät erlaubt werden und welche nicht. Diese *Policies* sind dabei nicht nur auf ein Endgerät beschränkt, sondern können auch auf anderen *Palm OS Cobalt* Endgeräten eingesetzt werden. Die komplette Sicherheitsfunktionalität wird von *RSA Security* entwickelt.

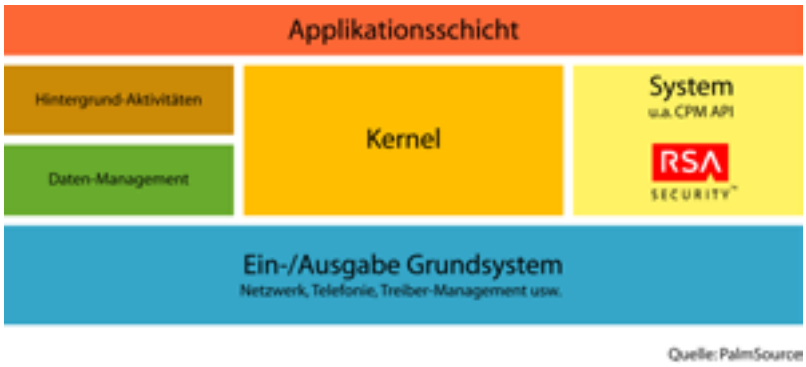


Abb. 2.16: *Aufbau von Palm OS Cobalt*

Palm OS Cobalt ist seit Februar 2004 veröffentlicht und liegt bereits in der Version 6.1 vor. Bis heute gibt es aber kein Endgerät in dem dieses neue Betriebssystem zum Einsatz kommt. Die Hersteller und Entwickler bevorzugen immer noch das *Palm OS Garnet* (Version 5.x). Wann Endgeräte mit dem neuen Betriebssystem ausgestattet werden ist noch

[9] *siehe* Kapitel 3.2.2 ab Seite 24
[10] *siehe* Kapitel 3.3.2 ab Seite 34

fraglich. Da aber *Palm OS Cobalt* weitgehend auf den Smartphone-Bereich ausgerichtet ist und dieser Markt momentan sehr stark wächst, ist es wahrscheinlich nur eine Frage der Zeit, bis die ersten Endgeräte mit diesem System auf den Markt kommen.

2.2.2.2 Symbian OS

Symbian OS ist die Weiterentwicklung von *EPOC*, das PDA-Betriebssystem von *Psion*, die seit 1998 von *Symbian Ltd.*, der u.a. *Nokia*, *Sony Ericsson* und *Motorola* angehören, vorangetrieben wird. *Smybian OS* wird heute ausschließlich in Smartphones eingesetzt und ist in diesem Sektor mit ca. 85% Marktführer[11]. Der *Nokia Communicator* oder das *Sony Ericsson P800/P900* sind die wohl bekanntesten Vertreter mit diesem Betriebssystem. Als *Series 60* kommt *Symbian OS* aber auch in Mobiltelefonen, die nicht primär als Smartphone Verwendung finden, wie z.B. das Spiele-Handy *Nokia N-Gage*, zum Einsatz.

Symbian OS ist vollständig multitasking- und multithreadingfähig und unterstützt ausschließlich ARM-Prozessoren. Die Aufgaben des Betriebssystems sind stark kommunikationsorientiert, was auch aus dem Schichtenaufbau von *Symbian OS* deutlich wird. Die Basis wird bei *Symbian OS*, neben dem Treibermodell für die Hardware und dem Dateisystem, vom Kernel gebildet. Besonderheit an *Symbian OS* ist die Abhängigkeit des Kernels von der eingesetzten Hardware. Aus diesem Grund übernimmt der Kernel Aufgaben eines *Hardware Abstraction Layer* (*HAL*), um Applikationen den Zugriff auf verschiedene Hardware zu ermöglichen. Als Hardware unterstützt *Symbian OS* alle gängigen Erweiterungskarten, Displays und Tastaturen, die für PDAs und Mobiltelefone erhältlich sind. Als Speichermedium dienen dem Betriebssystem neben *RAM* und *ROM* auch mobile Speicherkarten wie *CompactFlash*, *Secure Digital Memory Cards* (*SD-Cards*) und *Memory Sticks*. Neben dem Kern-System befindet sich die Telefonie-Unterstützung, die alle gängigen Dienste, die für Telefonie notwendig sind, zur Verfügung stellt. Dabei werden alle bekannten GSM-Dienste, wie *General Packet Radio Service* (*GPRS*) und *High Speed Circuit Switched Data* (*HSCSD*), in vollem Umfang unterstützt. Seit Version 7 unterstützt *Symbian OS* auch die neue Mobilfunktechnik *Universal Mobile Telecommunications System* (*UMTS*). In der Version 8 wurde die *UMTS*-Unterstützung weiter verbessert.

Symbian OS hat auf dieser Ebene auch eine Sicherheitsschicht implementiert, die Grundfunktionalitäten für die Vertraulichkeit und Integrität von Daten, sowie für Authentifizierung zur Verfügung stellt. Diese Sicherheitsschicht besteht vereinfacht dargestellt aus drei Modulen: das *Kryptographie-Modul*, das *Kryptographie Framework* und das *Zertifikat Management Modul*. Im *Kryptographie Modul* findet man verschiedene symmetrische und asymmetrische Kryptographie-Algorithmen, wie z.B. *DES*, *3DES* oder *RSA*, sowie einige Hash-Funktionen, wie z.B. *MD5*[12] oder *SHA-1*[13]. Das *Kryptographie Framework* bietet die Unterstützung von *X.509* Zertifikaten und die Authentifizierung durch die z.B. bei Mobilfunktelefonen bekannte *Personal Identification Number* (*PIN*). Das Framework verwaltet dabei, welche Hardwa-

[11] *siehe* http://www.symbian.com/press-office/2004/pr041111.html
[12] *siehe* Kapitel 3.2.1 ab Seite 23
[13] *siehe* Kapitel 3.2.2 ab Seite 24

re bzw. Applikation über eines dieser Mechanismen geschützt wird. Letztes Modul ist das *Zertifikat Management Modul* welches direkt am *Kryptographie Framework* hängt und die Verwaltung der darin gespeicherten Zertifikate übernimmt. [10]

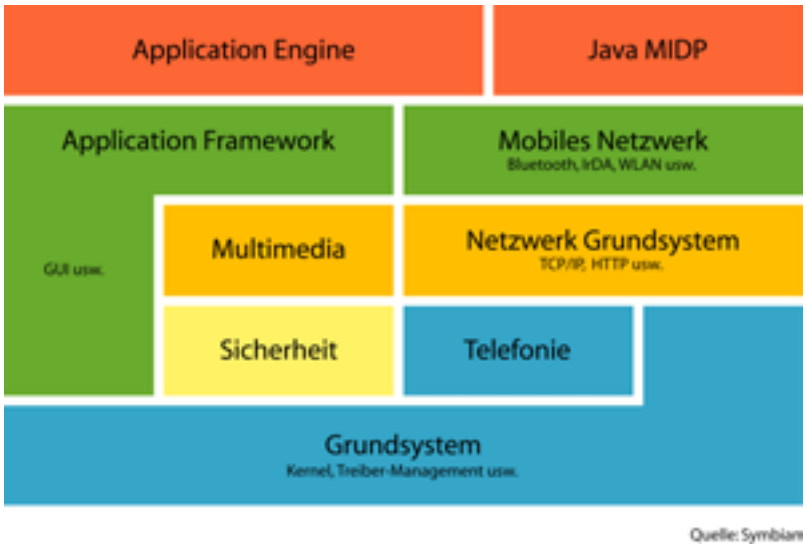


Abb. 2.17: *Aufbau von Symbian OS*

In der nächsten Ebene befindet sich die *Kommunikations-Infrastruktur*, die grundlegende Kommunikationsprotokolle wie *IPv4*, *HTTP* oder allgemein *TCP/IP* dem Betriebssystem zur Verfügung stellt. Als Schutz für eine gesicherte Datenübertragung unterstützt *Symbian OS* SSL und TLS. Aber auch mobile Übertragungstechniken wie das *Wireless Application Protocol* (*WAP*) werden von dieser Infrastruktur bereit gestellt. In dieser Ebene befindet sich ebenfalls die Multimedia-Schicht, die Funktionen z.B. für die Unterstützung einer Digitalkamera bietet.

Einen besonderen Stellenwert hat im *Symbian OS* das *Applikations Framework*, das teilweise direkt auf der Betriebssystem-Basis aufbaut und dadurch selbst grundlegende Funktionen beispielsweise für die graphische Benutzeroberfläche von *Symbian OS* bietet. Daneben verwendet es aber auch Dienste aus der Multimedia-Schicht und der Kommunikations-Infrastruktur und erweitert diese durch die Unterstützung von seriellen Schnittstellen, wie *USB* oder Funkstandards wie *Bluetooth* oder *IrDA*. Die Verwaltung der PIM-Daten ist sehr eng mit dem *Applikations Framework* verlochten und wird von sog. *Application Engines* erledigt. *Symbian* implementiert vollständig die *PersonalJava-Plattform*, erweitert durch *Mobile Information Device Profile* (*MIDP*), eine Sammlung von Java-Klassen speziell für mobile Endgeräte und *JavaPhone*, eine Sammlung von Java-Klassen für die Unterstützung von Telephonie. Dadurch ist es möglich Endgeräte mit *Symbian OS* durch selbst programmierte Java-Applikationen zu erweitern.

Die *Symbian OS* Smartphones *Nokia Communicator* und *Sony Ericsson P900* zeigen deutlich, wie die Zukunft von mobilen Endgeräten aussehen kann. Nicht zu kleine, aber handliche Endgeräte, die leicht zu bedienen sind und neben der Telefonie alle Möglichkeiten eines aktuellen PDAs bieten. Mobile Speicherkarten mit mehr als 128 MB und Funktech-

nologien wie Bluetooth und Wireless LAN machen sie zu mobilen Datenträgern. Die weite Verbreitung von *Symbian OS* wird nur langsam gestoppt. Nur *Windows CE*-basierte Endgeräte drängen langsam verstärkt auf den Markt, *Palm OS* findet, zumindest in Europa, nur sehr schwer seinen Platz im Markt für Smartphones.

Kapitel 3

Grundlagen für IT-Sicherheit

Das Betriebssystem alleine bietet in der Regel nicht genügend Schutz vor einem unerlaubten Zugriff auf das System oder die darauf befindlichen Daten. Aus diesem Grund gibt es einige Mechanismen, um einen Benutzer zu authentifizieren oder Systeme und Daten kryptografisch zu schützen. Generell unterscheidet man die drei Bereiche *Authentifizierung*, *Integrität* und *Vertraulichkeit*, die zusammen einen umfassenden Schutz für ein System bieten. Die *Authentifizierung* ermöglicht einem Benutzer seine Zugriffsberechtigung auf ein System zu verifizieren. Die *Integrität* stellt sicher, dass die Daten, mit denen man arbeiten will nicht manipuliert wurden. Die *Vertraulichkeit* schützt die Daten, meist durch ein kryptografisches Verschlüsselungsverfahren, vor unerlaubten Zugriff oder Verbreitung. Für alle Bereiche sind unterschiedliche Verfahren definiert und im Einsatz. Einige weit verbreitete Verfahren sollen im Folgenden in Ansätzen erklärt werden. Ein detaillierter Blick in alle Verfahren würde den Rahmen dieser Arbeit sprengen.

3.1 Zugriff durch Authentifizierung

Grundlegender Schutz für jedes System ist die *Authentifizierung* des Benutzers. Ist ein Benutzer gegenüber dem System nicht authentifiziert, so hat er zunächst keine Möglichkeit es zu benutzen. Dieser Schutz ist in vielen System standardmäßig implementiert und meist sogar als einziger Zugang vorausgesetzt. Die *Authentifizierung* kann über verschiedene Wege statt finden. Die Verfahren reichen von der Eingabe eines Authentifizierungscodes bis hin zu biometrischen Erkennungsmerkmalen.

3.1.1 Einfacher Passwortschutz

Der einfachste Weg der *Authentifizierung* ist die Vergabe eines Authentifizierungscodes. Diese Funktion ist in allen aktuellen Endgeräten, egal ob Laptop, PDA, Smartphone, Mobiltelefon

oder Desktop-PC, bereits fest implementiert und je nach Einstellung als „erforderlich“ oder „optional“ definiert. Selbst im *BIOS* von Laptops und Desktop-PCs besteht die Möglichkeit, einen Authentifizierungscode für das komplette System zu vergeben, durch den sich ein Benutzer bereits am Computer authentifizieren muss, bevor das eigentliche Betriebssystem gestartet wird. Die Bezeichnung für den Authentifizierungscode ist dabei von der jeweiligen Endgeräteart abhängig. Während man allgemein in der IT von einem *Passwort* spricht, wird bei Mobiltelefonen und Smartphones die Bezeichnung *PIN* verwendet. Eine *PIN* besteht dabei meist aus vier Ziffern, wobei ein *Passwort* eine Kombination aus Groß- und Kleinbuchstaben mit Ziffern und Sonderzeichen ist und eine variable Länge hat. Da *Passwort* in der IT die geläufigste Bezeichnung für einen Authentisierungscode ist, wird diese im Folgenden beibehalten.

Generell gesehen stellt ein *Passwort* einen sicheren Schutz für ein System dar. Ist dem Benutzer das Passwort nicht bekannt, so hat er auch keinen Zugriff auf das System. Wird zu dem Passwort zusätzlich ein Benutzername vergeben, dann erhöht sich die Sicherheit noch einmal, weil dann nur deren Kombination eine erfolgreiche *Authentifizierung* zulässt. Dabei ist der Schutz durch ein Passwort durch drei Faktoren gefährdet. Ein Faktor ist die Komplexität des Passworts, auch *Güte* des Passworts genannt. Besteht eine *PIN* z.b. aus der Ziffernfolge „0000“ oder „1234“ usw. so ist diese durch einfaches ausprobieren schnell zu erraten und das System bis zum nächsten Wechsel der *PIN* ungeschützt. Ebenso sieht es bei einem *Passwort* der Art „admin“ oder „abc“ aus. Von einem sicheren Passwort spricht man, wenn es mindestens eine Länge von acht Zeichen hat und aus einer Kombination von Groß- und Kleinbuchstaben mit mindestens einer Ziffer und/oder Sonderzeichen besteht. Diese Vorgaben sind durch die Administratoren der jeweiligen Systeme zu definieren. Zweiter Faktor ist die Lebensdauer eines Passwortes, auch *Password-aging* genannt. Wird eine Benutzername-Passwort-Kombination beispielsweise nur einmal vergeben und dann niemals geändert, so stellt dies ein gewisses Risiko dar. Ist die Kombination einmal gebrochen, so hat ein evt. Angreifer ständigen Zugriff auf das System mit den Rechten des jeweiligen Benutzers. Wenn davon sogar ein Administrator-Account betroffen ist, so könnte das für ein komplettes Netzwerk problematisch sein. Generell empfiehlt sich ein monatlicher Wechsel des Passworts, wobei die letzten zehn Passwörter nicht mehr benutzt werden dürfen. Alternativ ist auch der einmalige Wechsel des Passworts nach dem Anlegen möglich. Dies setzt aber voraus, dass auch der dritte Faktor, die Geheimhaltung des Passworts, eingehalten wird, was nur durch organisatorische Regeln möglich. Dabei werden die Benutzer angehalten ihr Passwort niemals weiter zu geben. Allgemein vergleichbar mit dieser Geheimhaltung ist die Verwendung der *PIN* bei einer Bankkarte. Diese *PIN* wird niemand öffentlich machen, weil man in Kombination mit der Bankkarte kompletten Zugriff auf das Konto und das darauf befindliche Geld hätte. Ebenso verhält es sich mit IT-Systemen. Die darauf befindlichen Daten sind in der Regel oft bares Geld wert. Deshalb muss den Mitarbeitern untersagt werden, Passwörter öffentlich zugänglich aufzuschreiben oder sogar an den Monitor zu kleben.

Inwiefern die drei Faktoren in einem Unternehmen umgesetzt werden, bleibt den jeweiligen Administratoren oder der Unternehmensleitung vorbehalten. Meist sind einige Faktoren praktisch nicht durchführbar, weil es zu viele Benutzer im Unternehmen gibt und die Einhaltung der Regeln evt. nicht kontrollierbar ist. Generell ist aber der Einsatz aller drei Faktoren zu empfehlen.

3.1.2 Token, Smartcards und Zwei-Faktor-Authentifizierung

Eine weitere Möglichkeit der *Authentifizierung* ist der Einsatz von hardwarebasierte Lösungen, sog. *Token* oder *Smartcards*. Bei beiden Verfahren wird grundsätzlich die selbe Technologie und Infrastruktur verwendet und nur das Aussehen der Hardware unterscheidet sich. *Smartcard*-Lösungen kommen im Sicherheitsumfeld von Unternehmen schon häufiger zum Einsatz. Sei es für den Zugang zu gesicherten Bereichen oder nur zur Zeiterfassung an zentralen Terminals. Diese *Smartcards* werden hier z.B. auch als Dienstausweis mit aufgedrucktem Bild usw. eingesetzt. Eine *Smartcard* enthält zusätzlich einen Chip, und gleichen im Aussehen einer Geld- oder Telefonkarte. Auf diesen Chip können zusätzliche Daten gespeichert werden und enthält oftmals auch diverse vorinstallierte Ver- und Entschlüsselungsfunktionen. Damit man eine *Smartcard* an einem Rechnersystem benutzen kann, ist ein spezielles Lesegerät nötig, das direkt über eine serielle Schnittstelle, wie z.B. USB, an den Laptop oder Desktop-PC angeschlossen wird. Für Laptops gibt es u.a. Lesegeräte für die *PCMCIA*-Schnittstelle. Zumindest für *Windows CE*-basierte PDAs gibt es mittlerweile auch Lesegeräte für die *CompactFlash*-Schnittstelle. Diese machen den PDA schon allein wegen der Größe der *Smartcards* unhandlich und sind für eine praktischen Einsatz kaum denkbar.

Abb. 3.1: *Smartcard als Dienstausweis*

Ein *Token* hat die Form eines USB-Sticks und kann ohne Zusatzgerät in eine vorhandene USB-Schnittstelle gesteckt werden. Der *Token* hat genauso wie die *Smartcard* einen Chip eingebaut, auf dem Daten gespeichert werden können und Funktionen zur Ver- und Entschlüsselung vorinstalliert sind. Diese *Tokens* können aber nur bei Laptops oder Desktop-PCs verwendet werden, da in aktuellen PDAs und Smartphones keine USB-Schnittstelle vorhanden ist. Welche Funktionen im Einzelnen eine *Smartcard* oder ein *Token* kann, hängt vom jeweiligen Hersteller ab.

Abb. 3.2: *USB-Token*

Da sich nur die Form der Hardware unterscheidet läuft die Authentifizierung bei beiden Lösungen gleich ab. Auf der *Smartcard* bzw. dem *Token* wird ein personenbezogenes Zertifikat verschlüsselt abgelegt. Das Zertifikat entspricht dabei einem gängigem Standard, wie z.B. X.509 oder PKCS#11 und besteht aus einem Paar mit einem öffentlichen und geheimen Schlüssel. Die Zertifikate werden durch eine *Registration Authority (RA)* bei einer zentralen Stelle, der *Certification Authority (CA)*, beantragt. Die *CA* stellt dann das Zertifikat zur Verfügung und signiert es digital. Die Signierung bestätigt die Echtheit des Zertifikats und liefert zustätzlich den Nachweis, dass der Benutzer der das Zertifikat benutzt auch der ist, für den er sich ausgibt. Die Zertifikate sind zusätzlich in einem Verzeichnisdienst, beispielsweise einem LDAP-Server, gespeichert und können dort über einen *Validierungsdienst* auf Gültigkeit geprüft werden. Dieser Verzeichnisdienst hält auch eine *Certification Revocation List (CRL)* bereit, in der alle abgelaufenen oder ungültigen Zertifikate aufgeführt sind. Diese Struktur wird auch als *Public Key Infrastructure (PKI)* bezeichnet. Stellt der Benutzer nun eine Verbindung zwischen

Smartcard oder *Token* und einem Laptop oder Desktop-PC her, um z.B. eine Betriebssystem-Anmeldung durchzuführen, dann wird über den *Validierungsdienst* die Gültigkeit des Zertifikats überprüft. Ist das Zertifikat gültig, so erhält der Benutzer Zugang zu dem System bei dem er sich authentifizieren will.

Abb. 3.3: *Beispielhafter Aufbau einer PKI*

Sollte *Token* oder *Smartcard* einmal gestohlen werden bzw. verloren gehen, so ist damit generell auch der Zugriff auf das System des jeweiligen Benutzers möglich. Um das zu verhindern, ist die Verwendung von *Token* bzw. *Smartcard* durch eine *Zwei-Faktor-Authentifizierung* zusätzlich geschützt. Bei einer *Zwei-Faktor-Authentifizierung* gibt es immer etwas, was man hat, beispielsweise einen *Token*, und etwas, was man weiß. Letzter Faktor ist meist eine vierstellige *PIN*, die natürlich geheim gehalten werden muss. Somit kann *Token* oder *Smartcard* ohne die dazugehörige *PIN* nicht benutzt werden. *Token* oder *Smartcard* können bei der *Zwei-Faktor-Authentifizierung* auch durch zufällig generierte *Einmal-Passwörter* ersetzt werden. Der Benutzer hat dazu einen *Token*, der nicht mit einem System verbunden werden muss, sondern nur die Generierung des *Einmal-Passworts* vornimmt. Dieser *Generator-Token* erzeugt nach einem Zufallsprinzip, das je Hersteller unterschiedlich ist, eine Zeichenfolge aus Ziffern und Buchstaben. Die Länge der Zeichenfolge ist wieder vom Hersteller abhängig, hat aber in der Regel mindestens sechs Stellen. Dieses *Einmal-Passwort* ist der erste Faktor für die *Authentifizierung*. Als zweiter Faktor kommt wieder eine vierstellige *PIN* dazu, die entweder vor die Zeichenfolge gestellt oder hinten angehängt wird. Auch das ist wieder herstellerabhänig. In Kombination mit einem Benutzernamen kann damit eine Authentifizierung nach dem *Zwei-Faktor-Verfahren* statt finden. Die Kombination aus Benutzername, *Einmal-Passwort* und *PIN* wird von einem *Authentisierungs-Server* auf Gültigkeit überprüft und bei Erfolg der Zugriff auf das System gewährt. Das *Zwei-Faktor-Verfahren* ist derzeit die sicherste Lösung für eine *Authentifizierung* und wird deswegen allgemein auch als *strong authentication* bezeichnet. Sollte der *Generator-Token* einmal verloren gehen oder gestohlen werden, so ist dieser unbrauchbar, wenn man nicht einen gültigen Benutzernamen und eine gültige *PIN* dazu besitzt. Der Einsatz einer *Zwei-Faktor-Authentifizierung* ist für Unternehmen mit einem erhöhten Sicherheitsbedarf dringend empfohlen.

Abb. 3.4: *Passwort-Generator SecurID von RSA Security*

3.1.3 Biometrische Verfahren

Nicht ganz neu ist die *Authentifizierung* anhand biometrischer Merkmale. Wissenschaftlich gesehen wurde im 19. Jahrhundert der Beweis erbracht, dass kein Fingerabdruck dem anderen gleicht, nicht einmal von eineiigen Zwillingen. Somit wurde erkannt, dass ein einzelner Fingerabdruch zur Identifiezierung einer Person verwendet werden kann. Noch heute wird das damals beschriebene Verfahren z.B. in der Kriminalistik eingesetzt. Erst seit ungefähr Ende der 1990er Jahre ist die Erkennung aber genau genug, um auch praktisch bei der *Authentifizierung* an IT-Systemen eingesetzt zu werden. Bei einem biometrischen Verfahren werden besondere Merkmale von Menschen analysiert und dadurch der Zugang zu einem System gesteuert. Ein Mensch hat eine Vielzahl an biometrischen Merkmalen, angefangen von der Handschrift bzw. Unterschrift, bis hin zu Körperabmaßen oder Merkmalen in Gesicht und Stimme. Auch eine Erkennung der DNA ist durchaus möglich. Durchgesetzt für den praktischen Einsatz haben sich aber die Erkennung von Iris bzw. Retina und von Fingerabdrücken, dem sog. Fingerlinienbild. Beide Verfahren werden schon seit längerem z.B. im Gebäudezugangsschutz eingesetzt. Um in einen besonders gesicherten Bereich zu kommen, wird zunächst ein oder mehrere biometrische Merkmale analyisiert und dann der Zugang geprüft. Ähnlich verhält es sich beim Zugang zu einem elektronischen System. Eine Software nimmt beispielsweise über eine Webcam die Iris bzw. Retina auf und vergleicht sie mit den gespeicherten Daten. Findet eine Übereinstimmung statt, so erhält der Benutzer Zugriff auf das System. Bei einem Fingerabdruck findet prinzipiell die gleiche Überprüfung statt. Hier wird ein oder mehrere ausgewählte Finger auf einen speziellen Fingerabdruckscanner gedrückt und eine Software sucht in ihrer Datenbank nach einer Übereinstimmung. Wird diese gefunden, so erhält der Benutzer wieder Zugriff auf das System.

Die Iris/Retina-Erkennung ist mit einem großen Aufwand verbunden und kommt eher selten zum Einsatz, für die Fingerabdruckerkennung sind aber bereits viele Lesegeräte am Markt. Der amerikanische Hersteller *Cherry* hat beispielsweise mehrere Tastaturen oder Mäuse im Sortiment, die genau diese Funktion übernehmen. Es gibt auch bereits Laptops, z.B. *Acer TravelMate 739TLV*, in denen ein solches Lesegerät bereits fest eingebaut ist. Und auch PDAs, wie z.B. *HP iPAQ hx2750 PocketPC*, haben einen Fingerabdruckleser. Die Erkennung ist sehr genau und kann auf jeden beliebigen Finger angewendet werden. Voraussetzung ist natürlich, dass man nichts an dem erfassten Finger verändert hat. Je nach Qualität des Lesegerätes reicht es manchmal einfach nur aus, wenn die Hände mal fettig sind oder der Fingerabdruck zu schnell oder zu ungenau auf dem Lesegerät platziert wurde. Die Technik wird ständig verbessert und liefert heute eine Erfolgsquote von ca. 95%.

Abb. 3.5: *Cherry FingerTIP ID Mouse M-4000*

Eine sicherere *Authentifizierung* als biometrische Merkmale gibt es derzeit nicht. Ein Mensch, und damit die benötigten Zugangsdaten, kann nicht einfach gestohlen oder nachgemacht werden. Jedoch sind viele Verfahren noch zu kompliziert und technisch kaum einsetzbar. Die Erkennung ist vielmals noch zu ungenau, dass sich ein praktischer Einsatz nicht

wirklich lohnt. Auch ist ein solches *Authentifizierungs*-Verfahren noch recht umstritten und findet kaum Zustimmung.

3.2 Überprüfung der Integrität

Um sicher zu stellen, dass z.B. Daten während einer Übertragung oder der letzten Benutzung nicht verändert wurden, ist es wichtig die *Integrität* dieser Daten zu überprüfen. Diese Überprüfung kann u.a. mit Hashing-Verfahren durchgeführt werden. Bei einem Hashing-Verfahren wird aus einer variablen Eingabegröße ein Wert aus konstanter Länge generiert. Dabei wird dieser Wert einmal vor der Übertragung ($= h$) und nochmal nach der Übertragung ($= h^{'}$) gebildet. Anschließend findet ein Vergleich der beiden Werte h und $h^{'}$ statt. Sind beide Werte gleich, dann kann man davon ausgehen, dass die Daten nicht verändert wurden. Jedoch kann das nicht generell für alle Hashing-Verfahren angenommmen werden. Zwei weit verbreitete Hashing-Verfahren werden nun kurz vorgestellt.

3.2.1 MD5

Das *Message Digest 5*, kurz *MD5*, ist wohl das bekannteste Hashing-Verfahren und kommt seit 1991 in vielen Anwendungen und Protokollen zum Einsatz. Als Eingabegröße akzeptiert *MD5* eine Nachricht von variabler Größe und erzeugt eine Ausgabe von konstant 128 Bit, die *Fingerabdruck* oder *Message Digest* genannt wird. Allgemein sprechen wir von der Ausgabe als *Hash-Wert*.

Die Eingabedaten werden bei *MD5* in Blöcke von jeweils 512 Bit aufgeteilt. Sollte ein Block nicht die Länge von 512 Bit haben, wird zunächst ein Bit mit dem Wert „1“, dann soviele Bits mit dem Wert „0“ angehängt, bis sich eine Gesamtlänge des Blockes von $512Bit - 64Bit = 448Bit$ ergibt. Die restlichen Bits werden durch eine 64 Bit lange Integerzahl abgebildet, die für die Gesamtlänge der Nachricht steht. Dieser Vorgang wird auch als *Padding* bezeichnet. Im nächsten Schritt werden vier Register von jeweils 32 Bit Länge definiert. Diese vier Register werden während der Berechnung des *Hash-Wertes* mehrfach verwendet. Die eigentliche Berechnung findet in vier Runden statt. Dazu wird in jeder Runde eine andere Funktion verwendet, die aus drei jeweils 32 Bit langen Eingabewerten einen 32 Bit langen Ausgabewert errechnet. Die vier Funktionen F, G, H und I lauten[1]:

$$\begin{aligned} F(X,Y,Z) &= (X \wedge Y) \vee (\neg X \wedge Z) \\ G(X,Y,Z) &= (X \wedge Z) \vee (Y \wedge \neg Z) \\ H(X,Y,Z) &= X \otimes Y \otimes Z \\ I(X,Y,Z) &= Y \otimes (X \vee \neg Z) \end{aligned}$$

Diese vier Funktionen werden auf alle 512 Bit-Blöcke durchgeführt. Die Ergebnisse von F, G, H und I werden zusammengesetzt und bilden letztendlich den 128 Bit *MD5 Hash-Wert*. Dies ist eine sehr vereinfachte Darstellung des *MD5*-Verfahrens. Eine genaue Erklärung ist unter [24] nachzulesen.

[1] $\otimes$ = XOR-Verknüpfung; $\wedge$ = AND; $\vee$ = OR; $\neg$ = NOT

MD5 galt bisher als sehr sicheres Verfahren für die Berechnung von *Hash-Werten*. Im August 2004 fand jedoch ein chinesisches Team von Wissenschaftlern durch eine Analyse-Methode eine Kollision auf das komplette *MD5*-Verfahren. Die Wissenschaftler suchten dabei zu einer bekannten Nachricht M eine Kollisionsnachricht $M^{'}$ die den gleichen *MD5 Hash-Wert* ergibt wie für die Nachricht M. Nach Angabe der Wissenschaftler hat die Suche nach der Nachricht $M^{'}$ auf einem *IBM P690 Cluster* nur eine Stunde gedauert. Der komplette Angriff auf das *MD5*-Verfahren wurde unter [29] veröffentlicht.

Dieser Angriff kann aber keine Rückschlüsse von einem *MD5 Hash-Wert* auf die Ausgangsdaten ziehen, sondern sucht nur nach einer alternativen Nachricht für den gleichen *Hash-Wert*. Ein Angriff auf Passwörter könnte dadurch noch recht sicher sein, wenn man nur den eigentlichen *Hash-Wert* des Passworts zur Verfügung hat. Auf lange Zeit werden sich aber bestimmt leistungsfähigere Methoden entwickeln, um das *MD5*-Verfahren weiter zu brechen. Die Verwendung von *MD5* wird daher nicht mehr empfohlen.

3.2.2 SHA

Besser geeignet ist der *Secure Hash Algorithm*, kurz *SHA*. Der *SHA* wurde Anfang der 1990er Jahre von der *National Security Agency* (*NSA*) und dem *National Institute ofe Standards and Technology* (*NIST*) entwickelt. Die heute meist verbreitete Version *SHA-1* wurde 1995 veröffentlicht und dient dazu von einer Nachricht oder allgemein von Daten einen *Hash-Wert* bzw. eine *Message Digest* zu erstellen. Bei *SHA-1* wird die Nachricht als Bitfolge verarbeitet, wobei die Länge der Nachricht gleich der Anzahl der Bits ist. Wenn die Länge der Nachricht durch 8 teilbar ist, so wird die Nachricht in hexadezimalen Zahlen dargestellt. Ziel ist es, ähnlich wie bei *MD5*, die Nachricht in Blöcke von 512 Bit Länge aufzuteilen. Der *SHA-1* erzeugt somit aus Nachrichten mit einer Länge von bis zu 2^{64} Bits einen *Hash-Wert* mit 160 Bits Länge. Er ist somit besser vor Kollissionen geschützt wie der *MD5*-Algorithmus.

Die Erzeugung dieses *Hash-Wertes* ähnelt sehr dem Verfahren, das bei *MD5* angewendet wird. Zuerst wird die Nachricht in die 512 Bit-Blöcke zerlegt. Ist ein Block wieder kürzer als 512 Bit, so wird zunächst ein Bit mit dem Wert „1“ und danach soviele Bits mit dem Wert „0“ angehängt, bis der Block eine Länge von $512Bits - 64Bits = 448Bits$ erreicht. Für die restlichen 64 Bit wird wieder eine Integerzahl eingesetzt, die die ursprüngliche Länge der Nachricht repräsentiert. Die Berechnung des *Hash-Wertes* wird in 80 Runden (t) vollzogen, in denen vier Funktionen und vier Konstanten, jeweils abhängig von der jeweiligen Runde, verwendet werden:

Die Eingabewerte X, Y und Z der vier Funktionen haben jeweils eine Länge von 32 Bit und die Funktion selbst ebenfalls einen Ausgabewert von 32 Bit. Die vier Funktionen lauten[2]:

Als Konstanten (K) werden die folgenden Werte verwendet, hier in einer hexadezimalen Darstellung:

Für die *Hash-Wert*-Berechnung werden zusätzlich die fünf Variablen A, B, C, D und E

[2] $\otimes$ = XOR-Verknüpfung; $\wedge$ = AND; $\vee$ = OR; $\neg$ = NOT

$$
\begin{array}{lcll}
F(t;X,Y,Z) & = & (X \wedge Y) \vee (\neg X \wedge Z) & (0 \leq t \leq 19) \\
F(t;X,Y,Z) & = & X \otimes Y \otimes Z & (20 \leq t \leq 39) \\
F(t;X,Y,Z) & = & (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z) & (40 \leq t \leq 59) \\
F(t;X,Y,Z) & = & X \otimes Y \otimes Z & (60 \leq t \leq 79)
\end{array}
$$

Tabelle 3.1: *Funktionen für die Hashwert-Berechung in SHA-1*

$$
\begin{array}{lcll}
K(t) & = & 5A\ 82\ 79\ 99 & (0 \leq t \leq 19) \\
K(t) & = & 6E\ D9\ EB\ A1 & (20 \leq t \leq 39) \\
K(t) & = & 8F\ 1B\ BC\ DC & (40 \leq t \leq 59) \\
K(t) & = & CA\ 62\ C1\ D6 & (60 \leq t \leq 79)
\end{array}
$$

Tabelle 3.2: *Initialisierung der vier Konstanten von SHA-1*

mit einer Länge von 32 Bits benötigt. Bei *SHA-1* werden diese Variablen wie folgt, wieder in hexadezimaler Darstellung, initialisiert:

$$
\begin{array}{lcl}
A & = & 67\ 45\ 23\ 01 \\
B & = & EF\ CD\ AB\ 89 \\
C & = & 98\ BA\ DC\ FE \\
D & = & 10\ 32\ 54\ 76 \\
E & = & C3\ D2\ E1\ F0
\end{array}
$$

Tabelle 3.3: *Initialisierung der Variablen von SHA-1*

Die Berechnung selbst kann nach zwei unterschiedlichen Methoden ablaufen, wobei jede Methode zum selben *Hash-Wert* führt. Bei beiden Methoden wird ein 512 Bit-Block in 16 Teile von 32 Bits unterteilt, sog. *Words*[3], und die Werte der Variablen in den Variablen $H0$, $H1$, $H2$, $H3$ und $H4$ zwischengespeichert. In beiden Methoden wird ebenso eine *Links-Shift-Operation* $SL^n(X)$ durchgeführt, die wie folgt definiert ist:

$$SL^n(X) = (X << n) \vee (X >> 32 - n)$$

Dabei werden bei der <<-Operation n Bits ganz links von X abgeschnitten und an das Ergebnis n Nullen angehängt. Bei der >>-Operation geschieht das gleiche nur von rechts aus.

Bei der ersten Methode zur *Hash-Wert*-Berechnung wird für alle Runden eine *TEMP*-Variable wie folgt berechnet:

$$TEMP = SL^5(A) + F(t;X,Y,Z) + E + W(t) + K(t)$$

Zusätzlich wird bei dieser Methode ab der Runde 16 noch folgende Berechnung für das aktuelle *Word* (W) durchgeführt:

[3]Im Algorithmus wird das jeweilige Word als W(t) bezeichnet

$$W(t) = SL^1(\,W(t-3) \otimes W(t-8) \otimes W(t-14) \otimes W(t-16))$$

In der zweiten Methode wird eine *AND*-Verknüpfung von der Rundenzahl t und einer Variablen *MASK* mit dem Wert 00 00 00 0F durchgeführt, so dass die Variable s wie folgt berechnet wird:

$$s = t \wedge MASK$$

Nun ergibt sich für die Berechnung der Variable *TEMP* folgende Formel:

$$TEMP = SL^5(A) + F(t; X, Y, Z) + E + W[s] + K(t)$$

Außerdem wird bei dieser Methode ab der Runde 16 folgende Berechnung für $W[s]$ durchgeführt:

$$W[s] = SL^1(\,W[(s+13) \wedge MASK] \otimes W[(s+8) \wedge MASK] \otimes W[(s+2) \wedge MASK] \otimes W[s])$$

Soweit die Unterschiede in den beiden Methoden. Bei beiden Methoden folgen nun weitere Berechnungen und Zuweisungen der Ausgangsvariablen

$$E = D;\ D = C;\ C = SL^{30}(B);\ B = A;\ A = TEMP$$

und zum Schluss noch die Verknüpfung mit den zwischengespeicherten Variablen, das wie folgt abläuft:

$$H0 = H0 + A;\ H1 = H1 + B;\ H2 = H2 + C;\ H3 = H3 + D;\ H4 = H4 + E$$

Die fünf Variablen $H0$, $H1$, $H2$, $H3$ und $H4$ repräsentieren genau in dieser Reihenfolgen den *SHA-1 Hash-Wert* und hat somit eine Länge von 160 Bits. Dies ist wieder nur eine sehr oberflächliche Betrachtung des *SHA-1*. Weitere Details sind unter [11] und [18] zu finden.

Seit 2002 sind drei weitere Varianten des *SHA* vom *NIST* veröffentlicht worden. Diese *SHA-256*, *SHA-384* und *SHA-512* genannten Algorithmen werden oft auch kurz als *SHA-2* bezeichnet. Der Unterschied zu *SHA-1* liegt eigentlich nur in der Länge des generierten *Hash-Wertes*, die je nach Algorithmus 256, 384 oder 512 Bit ist. Dafür werden zum einen mehr Konstanten (bei *SHA-256* 64 $K(t)$, bei *SHA-384* und *SHA-512* 80 $K(t)$) verwendet. Auf die Darstellung aller Konstanten wird an dieser Stelle verzichtet, diese sind vollständig

unter [20] nachzulesen. Des weiteren dienen nicht fünf sondern acht Variablen zur Darstellung des *Hash-Wertes*. Die Varianten *SHA-384* und *SHA-512* können im Gegensatz zu *SHA-1* und *SHA-256* mit Eingangswerten von bis zu 2^{128} Bits umgehen.

Die acht Variablen von *SHA-256* haben eine Länge von 32 Bit und werden wie folgt initialisiert:

$$\begin{array}{lcl} A & = & 6A\ 09\ E6\ 67 \\ B & = & BB\ 67\ AE\ 85 \\ C & = & 3C\ 6E\ F3\ 72 \\ D & = & A5\ 4F\ F5\ 3A \\ E & = & 51\ 0E\ 52\ 7F \\ F & = & 9B\ 05\ 68\ 8C \\ G & = & 1F\ 83\ D9\ AB \\ H & = & 5B\ E0\ CD\ 19 \end{array}$$

Tabelle 3.4: *Initialisierung der Variablen von SHA-256*

Im Gegensatz dazu haben die Variablen von *SHA-384* und *SHA-512* eine Länge von 64 Bit und werden wie folgt initialisiert:

$$\begin{array}{lcl} A & = & CB\ BB\ 9D\ 5D\ C1\ 05\ 9E\ D8 \\ B & = & 62\ 9A\ 29\ 2A\ 36\ 7C\ D5\ 07 \\ C & = & 91\ 59\ 01\ 5A\ 30\ 70\ DD\ 17 \\ D & = & 15\ 2F\ EC\ D8\ F7\ 0E\ 59\ 39 \\ E & = & 67\ 33\ 26\ 67\ FF\ C0\ 0B\ 31 \\ F & = & 8E\ B4\ 4A\ 87\ 68\ 58\ 15\ 11 \\ G & = & DB\ 0C\ 2E\ 0D\ 64\ F9\ 8F\ A7 \\ H & = & 47\ B5\ 48\ 1D\ BE\ FA\ 4F\ A4 \end{array}$$

Tabelle 3.5: *Initialisierung der Variablen von SHA-384*

$$\begin{array}{lcl} A & = & 6A\ 09\ E6\ 67\ F3\ BC\ C9\ 08 \\ B & = & BB\ 67\ AE\ 85\ 84\ CA\ A7\ 3B \\ C & = & 3C\ 6E\ F3\ 72\ FE\ 94\ F8\ 2B \\ D & = & A5\ 4F\ F5\ 3A\ 5F\ 1D\ 36\ F1 \\ E & = & 51\ 0E\ 52\ 7F\ AD\ E6\ 82\ D1 \\ F & = & 9B\ 05\ 68\ 8C\ 2B\ 3E\ 6C\ 1F \\ G & = & 1F\ 83\ D9\ AB\ FB\ 41\ BD\ 6B \\ H & = & 5B\ E0\ CD\ 19\ 13\ 7E\ 21\ 79 \end{array}$$

Tabelle 3.6: *Initialisierung der Variablen von SHA-512*

Für die Berechnung der Hash-Werte werden zusätzliche eine *Shift-Operation* S^n und eine *Right-Shift-Operation* SR^n definiert:

$$S^n(X) = X >> n$$

$$SR^n(X) = (X >> n) \vee (X << 32 - n)$$

Die Berechnung selbst wird mit sechs logischen Funktionen durchgeführt, die bei *SHA-256* ein 32-bit-Wort bzw. bei *SHA-384* und *SHA-512* ein 64-bit-Wort ergeben. Diese Funktionen sind wie folgt definiert:

$$\begin{array}{rcl} F(X,Y,Z) & = & (X \wedge Y) \oplus (\neg X \wedge Z) \\ G(X,Y,Z) & = & (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z) \\ \sum_0^{256}(x) & = & SR^2(x) \oplus SR^{13}(x) \oplus SR^{22}(x) \\ \sum_1^{256}(x) & = & SR^6(x) \oplus SR^{11}(x) \oplus SR^{25}(x) \\ \sigma_0^{256}(x) & = & SR^7(x) \oplus SR^{18}(x) \oplus S^3(x) \\ \sigma_1^{256}(x) & = & SR^{17}(x) \oplus SR^{19}(x) \oplus S^{10}(x) \end{array}$$

Tabelle 3.7: *Funktionen für die Hashwert-Berechung in SHA-256*

$$\begin{array}{rcl} F(X,Y,Z) & = & (X \wedge Y) \oplus (\neg X \wedge Z) \\ G(X,Y,Z) & = & (X \wedge Y) \oplus (X \wedge Z) \oplus (Y \wedge Z) \\ \sum_0^{512}(x) & = & SR^{28}(x) \oplus SR^{34}(x) \oplus SR^{39}(x) \\ \sum_1^{512}(x) & = & SR^{14}(x) \oplus SR^{18}(x) \oplus SR^{41}(x) \\ \sigma_0^{512}(x) & = & SR^1(x) \oplus SR^8(x) \oplus S^7(x) \\ \sigma_1^{512}(x) & = & SR^{19}(x) \oplus SR^{61}(x) \oplus S^6(x) \end{array}$$

Tabelle 3.8: *Funktionen für die Hashwert-Berechung in SHA-384 und SHA-512*

Für die Berechnung des Hash-Wertes werden, anders als die obigen Formeln, rundenabhängig (t) einzelne *Words* (W) wie folgt berechnet[4]:

$$W_t = \begin{cases} M_t^i & 0 \le t \le 15 \\ \sigma_1^s(W_{t-2}) + W_{t-7} + \sigma_0^s(W_{t-15}) + W_{t-16} & 16 \le t \le t_{max} \end{cases}$$

Ähnlich wie bei *SHA-1* werden nun die Ausgangsvariablen A, B, C, D, E, F, G und H in den acht Hilfsvariablen $H0$, $H1$, $H2$, $H3$, $H4$, $H5$, $H6$ und $H7$ zwischengespeichert. Diese werden am Schluß noch einmal benötigt. Im Gegensatz zu *SHA-1* werden bei *SHA-2* zwei temporäre Variablen berechnet:

$$TEMP1 = H + \sum_1^s(E) + F(E,F,G) + K_t^s + W_t$$

$$TEMP2 = \sum_0^s(A) + G(A,B,C)$$

Die restlichen Werte werden in jeder Runde wie folgt definiert:

[4] t = Rundennummer; s = 256 für *SHA-256* bzw. 512 für *SHA-384/SHA-512*; t_{max} = 63 für *SHA-256* bzw. 79 für *SHA-384/SHA-512*;

$H = G;\ G = F;\ F = E;\ E = D + TEMP1;\ D = C;\ C = B;\ B = A;\ A = TEMP1 + TEMP2$

Zum Schluß werden wieder die zwischengespeicherten mit den berechneten Variablen kombiniert und es ergibt sich:

$$H0 = H0 + A;\ H1 = H1 + B;\ H2 = H2 + C;\ H3 = H3 + D;\ H4 = H4 + E;\ H5 = H5 + F;\ H6 = H6 + G;\ H7 = H7 + H$$

Die acht Variablen $H0$, $H1$, $H2$, $H3$, $H4$, $H5$, $H6$ und $H7$ haben je nach Algorithmus jeweils eine Größe von 32 Bit (*SHA-256*) bzw. 64 Bit (*SHA-384 / SHA-512*) und repräsentieren genau in dieser Reihenfolgen den Hash-Wert des jeweiligen Algorithmus. Der Hash-Wert hat somit eine Länge von 256 Bit (*SHA-256*) bzw. 512 Bit (*SHA-512*). Da bei *SHA-384* jede Variable eine Länge von 64 Bit hat, werden nur die ersten fünf Variablen verwendet, um den Hash-Wert mit der Länge von 384 Bit darzustellen.

Alle Einzelheiten zu *SHA-1* und den *SHA-2*-Varianten sind nochmals ausführlich in [20] dargestellt. Dort sind auch alle Konstanten definiert, die noch für die Berechnung der Hash-Werte benötigt werden.

SHA gilt heute als sicherstes Verfahren zur Berechnung von *Hash-Werten*. Aufgrund der großen Länge des *Hash-Wertes* ist es sicherer vor Kollisionen als z.B. *MD5*. Das *Bundesamt für Sicherheit in der Informationstechnik* (*BSI*) definiert *SHA* als Standard-Hashing-Verfahren für die Überprüfung der *Integrität* von Daten und Nachrichten. Ein Einsatz von *SHA-1*, besser noch *SHA-2*, ist deswegen dringend zu empfehlen.

3.3 Verfahren für Verschlüsselung

Als letzte Möglichkeit dient noch die eigentliche *Verschlüsselung* von Daten oder dem Datentransfer. Generell werden dabei die Daten für andere unlesbar abgespeichert. Versucht nun ein Angreifer die Daten auszulesen, so kann er mit diesen nichts anfangen, weil er keine Möglichkeit hat, die Daten zu entschlüsseln. Bei den Verfahren zur *Verschlüsselung* werden drei Arten unterschieden: die *symmetrische Verschlüsselung*, die *asymmetrische Verschlüsselung* und die *Hybrid-Verschlüsselung*. Bei der *Hybrid-Verschlüsselung* handelt es sich um die Kombination aus *symmetrischer* und *asymmterischer Verschlüsselung*. Je ein Beispiel für eine *symmetrische* und eine *asymmetrische Verschlüsselung* sollen im Folgenden etwas näher betrachtet werden. Da diese Verfahren sehr komplexe mathematische Verfahren darstellen, werden nur die wesentlichen Teile betrachtet. Eine detaillierte Erklärung der beiden Verfahren ist für diese Diplomarbeit nicht notwendig.

3.3.1 Symmetrische Verschlüsselung mit AES

Der *Advanced Encryption Standard* (*AES*) wird seit Oktober 2000 als Nachfolger des *Data Encryption Standard* (*DES*) angesehen. *DES* gilt seit 1998 als unsicher, da dieses Verfahren sehr leicht über eine *Brute-Force-Attacke* zu brechen sind. Somit rief die *NIST* Anfang 1997 zu einem Wettbewerb auf, der einen neuen symmetrischen Algorithmus finden soll. Dieser Algorithmus soll sowohl in Hardware als auch in Software umsetzbar, performant und vor allem widerstandsfähig gegen alle bekannten Methoden der Kryptoanalyse sein. Im April 1999 wurden von der *NIST* fünf Algorithmen (*MARS*, *RC6*, *Rijndael*, *Serpent* und *Twofish*) als brauchbar eingestuft und in die Endrunde geschickt. Nach diversen Analysen und öffentlichen Diskussion wurde im Oktober 2000 der belgische *Rijndael*-Algorithmus als Sieger gekürt und der *AES* war geboren.

AES ist ein sog. Blockchiffre. Die Blocklänge kann genauso wie die Schlüssellänge unabhängig voneinander 128, 192 oder 256 Bit erhalten. Jeder Block wird zunächst in eine zweidimensionale Tabelle mit vier Zeilen geschrieben. Die einzelnen Zellen haben dabei eine Größe von einem Byte und die Anzahl der Spalten ergibt sich aus der gewählten Blocklänge. Die Abmaße der Tabelle reichen somit von 4 Spalten bei 128 Bit bis hin zu 8 Spalten bei 256 Bit. Durch verschiedene Transformationen wird dann jeder Block mit verschiedenen Teilen des Schlüssels nacheinander verschlüsselt. Dieses „nacheinander“ wird im *AES* auch als „Runde“ (r) bezeichnet, welche von der Schlüssellänge (k) und der Blocklänge (b) abhängig ist. Aus all diesen Werten kann man folgende Tabelle aufstellen:

r	**b = 128 Bit**	**b = 192 Bit**	**b = 256 Bit**
k = 128 Bit	10	12	14
k = 192 Bit	12	12	14
k = 256 Bit	14	14	14

Tabelle 3.9: *Rundenlänge*

Zur Verschlüsselung verwendet *AES* eine Substitutionsbox, die als Basis für eine monoalphabetische Verschlüsselung[5] verwendet wird, deren mathematischer Zusammenhang fest im Algorithmus implementiert ist. Die Substitutionsbox gibt an, welches Byte wie getauscht wird und arbeitet somit byteweise.

Der *Rijndael*-Algorithmus teilt als erstes den Schlüssel in $r + 1$ Teilschlüssel auf, die auch *Rundenschlüssel* genannt werden. Mit diesen *Rundenschlüsseln* werden dann nach und nach die Daten verschlüsselt. Da die *Rundenschlüssel* die gleiche Länge wie die Blöcke haben müssen, wird die Länge über die Formel $b \cdot (r+1)$ expandiert. Dieser erste Schlüssel wird auch *Benutzerschlüssel* genannt und wird in der ersten Runde zur Verschlüsselung verwendet.

Die weiteren *Rundenschlüssel* werden aufgrund dieses *Benutzerschlüssels* erzeugt. Hierbei wird der *Benutzerschlüssel* zunächst in Nk 4-Byte Wörter[6] aufgeteilt. Nk berechnet sich aufgrund der Schlüssellänge (128 Bit $\rightarrow$ $Nk = 4$, 192 Bit $\rightarrow$ $Nk = 6$, 256 Bit $\rightarrow$ $Nk = 8$)

[5] Jedem Wert wird ein anderer Wert gegenübergestellt mit dem er codiert wird, z.B. $A \widehat{=} Z$, $B \widehat{=} Y$, $C \widehat{=} X$ usw.

[6] 1 Wort $\widehat{=}$ 32 Bit oder 8 Byte

und stellt die Anzahl der Spalten der geheimen Schlüsselmatrix dar. Somit ergeben sich die ersten Wörter $k_0, \ldots, k_{Nk-1}$. Das jeweils folgende Wort wird durch eine XOR-Verknüpfung des vorhergehenden Wortes k_{i-1} mit k_{i-Nk} erzeugt:

$$k_i = k_{i-1} \oplus k_{Nk-1}$$

Sollte der Fall auftreten, dass $i \equiv 0 \mod Nk$ ist, dann wird vor der Verknüpfung auf das Wort k_{i-1} die Funktion $F_{NK(k,i)}$ angewandt:

$$F_{NK(k,i)} := S(r(k)) \oplus c$$

Diese Funktion besteht aus einer linkszyklischen Verschiebung um ein Byte ($r(k)$), einer Ersetzung aus der *S-Box* aus dem *Rijndael*-Algorithmus ($S(r(k))$) und einer XOR-Verknüpfung dessen mit einer Konstanten c. Hat der Benutzerschlüssel eine Länge von 256 Bit ($Nk = 8$) kommt noch als Besonderheit hinzu, dass eine zusätzliche Substitution bei $i \equiv 4 \mod Nk$ eingeschoben wird. Wurden nun die benötigten $Nk \cdot (r + 1)$ Wörter erzeugt, können diese zu den Rundenschlüsseln zusammengesetzt werden und zur Ver- und Entschlüsselung verwendet werden. [30]

Um den Ablauf der Verschlüsselung besser darzustellen ist es sinnvoll die einzelnen Abläufe in vier Funktionen zusammenzufassen:

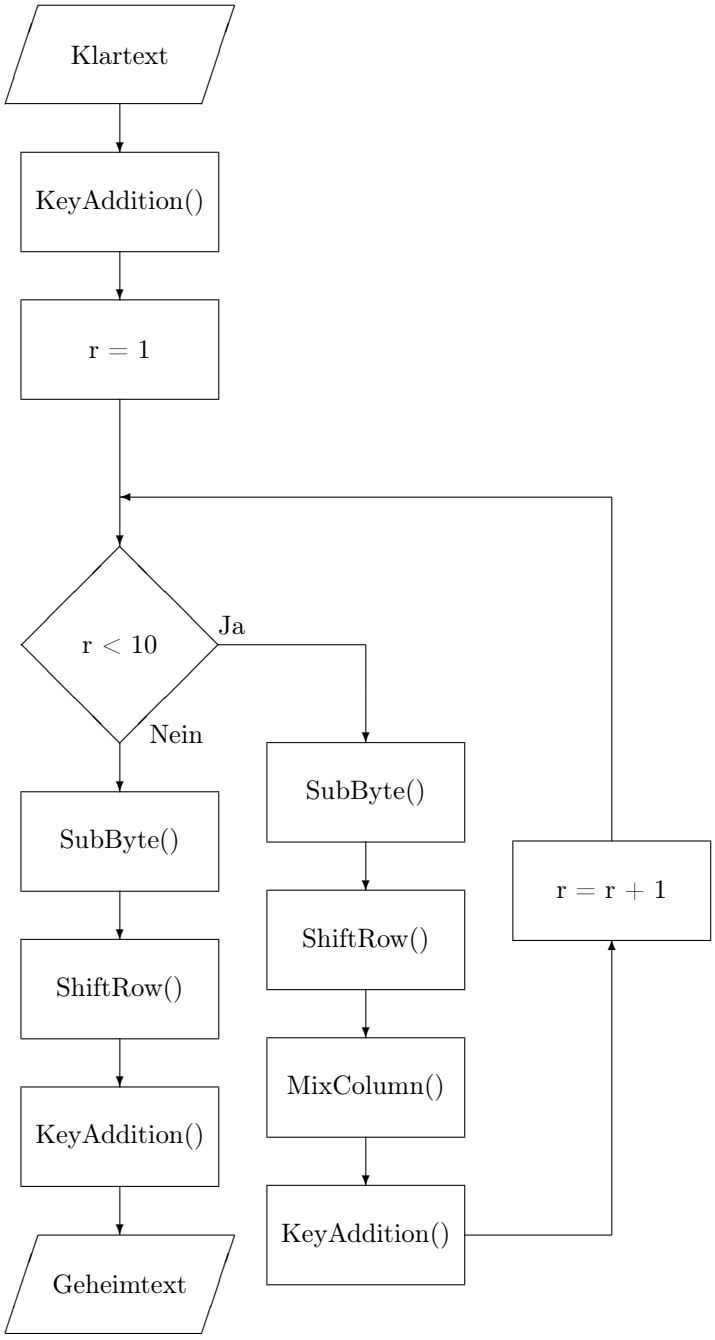
1. SubByte
   Hier findet eine Transformation in Form einer nicht-lineare byteweisen Substitution statt. Hierfür wird eine spezielle Substitutionsmatrix verwendet, die *S-Box*, mit 256 Einträgen mit einer Größe von je einem Byte. Somit bietet die *S-Box* unabhängig von der Schlüssellänge genügend Ersatzwerte. Wie diese *S-Box* aufgebaut wird, ist in [23] beschrieben. Jeder Block wird hierbei durch einen Wert aus der *S-Box* ersetzt. Die Entschlüsselung erfolgt durch eine inverse *S-Box*.
2. ShiftRow
   Jeder Block liegt in Form einer zweidimensionalen Tabelle mit vier Zeilen vor. In diesem Schritt werden Zeilen um eine bestimmte Anzahl von Spalten nach links verschoben. Überlaufende Zellen werden von rechts fortgesetzt. Die Anzahl der Verschiebungen ist wieder abhängig von der Blocklänge `b` und von der Zeile in der man sich gerade befindet:
3. MixColumn
   Nun werden die Spalten miteinander vermischt wobei zunächst jede Zelle einer Spalte mit einer Konstanten[7] multipliziert und dann mit XOR verknüpft.
4. KeyAddition
   In dieser Funktion wird eine bitweise XOR-Verschlüsselung zwischen dem aktuellen Block und dem aktuellen *Rundenschlüssel* vorgenommen. Dies ist die einzige Funktion in AES, die den Algorithmus vom *Benutzerschlüssel* abhängig macht.

[7] Werte für die Konstante: Zeile 1: 2, Zeile 2: 3, Zeile 3: 1, Zeile 4: 1

r	b = 128	b = 192	b = 256
Zeile 0	0	0	0
Zeile 1	1	1	1
Zeile 2	2	2	2
Zeile 3	3	3	4

Tabelle 3.10: *Verschiebung der Zeilen*

Der kompletten Ablauf des *Rijndael*-Algorithmus kann wie folgt dargestellt werden:


Klartext
KeyAddition()
r = 1
r < 10
Ja
Nein
SubByte()
ShiftRow()
KeyAddition()
Geheimtext
SubByte()
ShiftRow()
MixColumn()
KeyAddition()
r = r + 1

Die vollständige Beschreibung des *Rijndael*-Algorithmus und von *AES* ist unter [19] nachzulesen. *AES* hat sich sehr schnell als Standard-Verfahren für *Verschlüsselung* durchgesetzt und gilt bis heute als sicher. Viele neue Anwendungen und Protokolle verwenden ausschließlich *AES* zur Verschlüsselung oder unterstützen es auf alle Fälle. Ein Einsatz von *AES* für eine *symmetrische Verschlüsselung* ist nur zu empfehlen.

3.3.2 Asymmetrische Kryptographie mit RSA

Um nach der *RSA*-Methode[8] verschlüsseln zu können ist zunächst ein Paar aus öffentlichen und privaten Schlüssel notwendig. Dabei werden zwei frei gewählte große Primzahlen p und q ausgewählt, wobei $p \neq q$ gelten muss. Mit diesen Primzahlen läßt sich $N = p \cdot q$ berechnen. Anschließend wird folgende Berechnung durchgeführt:

$$\phi(N) = (p-1) \cdot (q-1) \text{ } (\phi = \text{Eulersche Funktion})$$

Danach wird eine natürlich Zahl $e > 1$ ausgewählt, die nicht mit $\phi(N)$ teilbar ist. Abschließend wird eine natürliche Zahl d berechnet, für die $e \cdot d \equiv 1 \mod \phi(N)$ gilt. Aus diesen Berechnungen werden nun die Zahlen N und e als öffentlicher Schlüssel verwendet und d, p und q und damit auch $\phi(N)$ als privater Schlüssel.

Zur Verschlüsselung einer Nachricht im Klartext (K) zu einem Geheimtext (G) verwendet der Algorithmus folgende Formel:

$$G \equiv K^e \mod N$$

Somit wurde der Klartext mit dem öffentlichen Schlüssel verschlüsselt.

Zum Entschlüsseln der Nachricht wird der private Schlüssel benötigt. Dazu wird nun die Zahl `d` als Exponent verwendet und über die diskrete Exponentialfunktion die Nachricht entschlüsselt. Der Nachrichtenempfänger verwendet dazu die Formel:

$$K \equiv G^d \mod N$$

RSA ist mit den Gesetzesanforderungen für Digitale Signaturen konform und ist dafür auch für den Einsatz im geschäftlichen Alltag geeignet.

[8] *RSA* steht für die Anfangsbuchstaben der Nachnamen der Entwickler Ronald L. **R**ivest, Adi **S**hamir und Lenoard **A**dleman

3.4 Praktischer Einsatz bei mobilen Endgeräten

In mobilen Endgeräten sind die hier beschriebenen Verfahren nur sehr eingeschränkt umgesetzt. Zwar besteht bei jedem Endgerät die Möglichkeit einen einfachen Passwortschutz zu aktivieren, aber Funktionen zur Verschlüsselung der Daten auf dem Endgerät fehlen zum Teil vollkommen. Erst nach und nach binden die Hersteller diese Möglichkeiten in ihre Betriebssysteme ein (*siehe* Kapitel 2.2.2.1 auf Seite 14 oder Kapitel 2.2.2.2 auf Seite 15. Außerdem ist die Unterstützung eines gesicherten Verbindungsaufbaus über einen *VPN*-Tunnel oder *SSL* bzw. *TLS* ebenfalls standardmäßig implementiert. Für eine *Zwei-Faktor-Authentifizierung* oder für die *Verschlüsselung* der Daten muss eine zusätzliche Software eingesetzt werden. Diese Software sollte dann auch die Überprüfung der *Integrität* ermöglichen, was auch nur sehr eingeschränkt möglich ist. Welche Lösungen vor allem für Unternehmen für diese Aufgaben geeignet sind zeigt das Kapitel 5.2.2 auf Seite 60.

Kapitel 4

Anbindung mobiler Endgeräte

Die Anbindung von mobilen Endgeräten an Netzwerken kann in verschiedener Weise realisiert werden. Dabei wird die Verbindung hauptsächlich zur Synchronisation der Daten und zur Benutzung verschiedener Applikationen, wie z.B. Telefonie oder Abrufen von Inhalten aus dem Internet verwendet. Neben der klassischen kabelgebundenen Verbindung existieren auch kabellose Verbindungen, die mit ein Vorteil der mobilen Endgeräte sind. Je nach Endgerätetyp sind dabei ebenfalls unterschiedliche Anbindungstypen vorhanden. Als Netzwerkprotokoll hat sich sowohl bei den kabelgebundenen, als auch bei den kabellosen Verbindungen TCP/IP durchgesetzt.

4.1 kabelgebundene Verbindung

Bei einer kabelgebundenen Verbinung ist das mobile Endgerät physikalisch mit einem Netzwerk oder Host-Rechner verbunden und stellt damit die einfachste Anbindung dar. Je nach Endgerätetyp werden hierbei unterschiedliche Kabelarten verwendet. Um beispielsweise einen Laptop mit einem Netzwerk zu verbinden wird ein *Twisted Pair-Ethernet*-Kabel verwendet, das direkt mit der Netzwerkkarte des Laptops und beispielsweise einem Router oder Switch des Netzwerkes verbunden wird. Somit ist der Laptop teil dieses Netzwerkes und kann auf alle Bereich zugreifen, auf die Berechtigungen bestehen. Auch bei PDAs oder Smartphones ist der Einsatz von Ethernet-Kabeln möglich, was aber nur über CompactFlash-Erweiterungskarten ermöglicht wird. Dabei sind Übertragungsraten bis 1 GBit/s möglich. Bei PDAs, Smartphones und Mobiltelefonen wird zur Anbindung in der Regel ein serieller Anschluss, entweder RS232[1], heutzutage aber eher USB, des Host-Rechners verwendet. Eine direkte Verbindung mit dem Netzwerk, wie etwa bei Ethernet, ist mit dieser Verbindungart allerdings nicht möglich, sondern wird stets über den Host-Rechner durchgeführt. Für PDAs gibt es sehr oft auch eine Dockingstation, in der das Endgerät während der Synchronisation abgestellt werden kann. Der Host-Rechner dient durch die installierte *HotSync*- oder *ActiveSync*-Software als eine Art *Router*, die dem mobilen Endgerät den Zugang zu Netzwerk oder Internet er-

[1] identifiziert sich bei Computern mit *Microsoft Windows* als *COM1*- oder *COM2*-Schnittstelle

möglicht. Mit der aktuellen USB Version 2.0 erreicht man eine Datenrate von bis zu 480 MBit/s.

4.2 kabellose Verbindung

4.2.1 IrDA

Der Infrarot-Anschluß ist in jedem mobilen Endgerät standardmäßig als „Funkschnittstelle“ vorhanden und wird bereits seit 1993 eingesetzt. Beschrieben ist die physikalische Spezifikation und der Standard für die Kommunikation über Infrarot von der *IrDA*. Eine Infrarot-Verbindung ist allerdings nur über die sehr kurze Distanz von maximal 1 m möglich und die beiden kommunizierenden Schnittstellen müssen in direkter Sichtverbindung stehen. D.h. die Schnittstellen müssen direkt aufeinander ausgerichtet sein, damit ein Endgerät ein anderes Endgerät erkennt. Dadurch ist eine gewisse Abhörsicherheit gegeben. Die Datenrate bei der aktuellen *IrDA* Version 1.1 reicht von 4MBit/s (Fast-Infrared) bis zu 16MBit/s (Very-Fast-Infrared) und ist somit für den Datenaustausch größerer Dateien eher ungeeignet. Für eine Synchronisation von PIM-Daten ist die Geschwindigkeit aber durchaus ausreichend.

4.2.2 Bluetooth

Die mobilen Endgeräte werden zusätzlich immer mehr durch den *Bluetooth*-Standard erweitert. *Bluetooth* wurde 1994 von dem Mobilfunkhersteller *Ericsson* entwickelt, um Mobiltelefone kabellos mit Zusatzgeräten zu verbinden. Die Entwicklung wird seit 1998 von der *Bluetooth SIG* weitergeführt, der neben *Ericsson* Firmen wie *Intel*, *Nokia*, *Toshiba*, *3Com*, *Microsoft* und *Motorola* angehören. Seit 2001 wird *Bluetooth* auf Basis der Version 1.1 sogar als offiziell als *IEEE 802.15.1* Standard für *Wireless Personal Area Network* definiert. *Bluetooth* wird aber nicht nur in mobilen Endgeräten, sondern auch in Computer-Peripherie wie z.B. Druckern, Eingabegeräte oder Audiogeräte eingesetzt. Damit die Kommunikation zwischen den einzelnen *Bluetooth*-Endgeräten ohne Konfiguration funktioniert, sind verschiedene Profile definiert, die eine spezielle Aufgabe beschreiben. So gibt es z.B. Profile für den Datentransfer, für Telephonie, für Eingabegeräte, für serielle Verbindungen uvm. Derzeit gibt es in der aktuellen Version *Bluetooth 1.2* ca. 60 unterschiedliche Profile.

Bei vielen Mobiltelefonen, vor allem bei Smartphones, ist *Bluetooth* mittlerweile Standard. Auch in aktuellen PDAs ist *Bluetooth* bereits eingebaut. Sollte das nicht der Fall sein, so existiert eine Vielzahl an Erweiterungskarten für die *CompactFlash*- bzw. *SDIO*[2]-

[2] **S**ecure **D**igital **I**nput *O*utput

Schnittstelle. Für Laptops, aber auch Desktop-PCs, gibt es USB-Adapter, die den Rechner um die *Bluetooth*-Unterstützung erweitert. Es ist jedoch darauf zu achten, welche Profile der jeweilige USB-Adapter unterstützt und somit für das Endgerät zur Verfügung stellt. Ist die Unterstützung ausreichend, so ist es z.B. auch möglich ein Mobiltelefon als Maus für den Laptop zu benutzen und beispielsweise PowerPoint-Präsentationen über diese mobile Endgerät zu steuern. Den Einsatzmöglichkeiten von *Bluetooth* sind somit kaum Grenzen gesetzt, vorausgesetzt die beiden Kommunikationspartner unterstützen das gleiche Profil.

Abb. 4.1: *Bluetooth USB-Dongle von Belkin*

Bluetooth erlaubt bis zu 255 Teilnehmer in einem Netzverbund, das *Piconet* genannt wird, wobei maximal acht Teilnehmer gleichzeitig aktiv sein dürfen. Einer dieser acht aktiven Teilnehmer übernimmt dabei eine Master-Funktion und verteilt die Anfragen an die anderen, die als *Slaves* bezeichnet werden. Bis zu zehn *Piconet*s kann man zu einem *Scatternet* zusammenfassen, in dem jedes *Piconet* durch eine unterschiedliche Frequenz-Hopping-Folge identifiziert wird. Dabei wechselt die Frequenz bis zu 1.600 mal pro Sekunde durch 79 Kanäle im 2.4 GHz Frenquenzband. Die Teilnehmer in einem *Scatternet* können miteinander in Kontakt treten. Die Reichweite eines *Bluetooth*-Endgeräts ist dabei ganz davon abhängig welcher Klasse es zugeordnet ist. Im aktuellen *Bluetooth*-Standard wird derzeit zwischen drei Klassen unterschieden, wobei Klasse I eine Reichweite von 100m, Klasse II eine Reichweite von 50m und Klasse III eine Reichweite von 10m hat. In Laborbedingungen mit einem speziell modifizierten *Bluetooth*-Modul und einer Richtfunkantenne wurde aber auch schon eine Reichweite von ca. 1,74 km erreicht[3].

Die eigentliche Kommunikation zwischen zwei *Bluetooth*-Endgeräten, z.B. ein Laptop mit einem Mobiltelefon, erfolgt dann in zwei Stufen. Zunächst werden die beiden *Bluetooth*-Endgeräte durch ein Handshake-Protokoll miteinander verbunden. Bei diesem sog. *Pairing* oder *paarweise verbinden* wird eine beliebige Endgeräte-PIN definiert, die beiden Kommunikationspartnern bekannt sein muss. Wurde an beiden stellen die PIN eingegeben und akzeptiert, wird ein 128bit langer *Combination Key* erzeugt und in den Endgeräten abgespeichert. Es besteht dann eine permanente Verbindung zwischen den beiden Endgeräten und ein erneutes *Pairing* ist für spätere Verbindungen in der Regel nicht mehr nötig. Synchronisiert ein Mobiltelefon z.B. immer mit dem gleichen Host-Rechner, so stimmt das. Benutzt man den gleichen *Bluetooth* USB Adapter an verschiedenen Host-Rechner z.B. an einem Desktop und einem Laptop, so kann es vorkommen, dass dennoch ein erneutes *Pairing* notwenig ist, obwohl Dektop und Laptop dem Mobiltelefon bekannt sind. Im zweiten Schritt wird ein gemeinsam vorhandenes Profil ausgewählt über das die weitere Kommunikation statt findet.

Die Identifikation der einzelnen Teilnehmer erfolgt dann anhand der weltweit eindeutigen, 48bit langen MAC-Adresse, die hier auch *Bluetooth Device Address* genannt wird. Das Endgerät, das die Verbindung als erstes zur Verfügung stellt ist dabei das Master-Endgerät. Die Datenübertragung findet entweder unverschlüsselt oder verschlüsselt statt finden. Die Verschlüsselung ist in zwei Stufen aufgeteilt. In der ersten Stufe steht die *Link Level Security*, bei der die Verschlüsselung schon vor dem eigentlich *Pairing* statt findet. Die zweite Stufe ist die

[3] *siehe* http://www.heise.de/newsticker/meldung/49907

Service Level Security, bei der je nach benutztem Dienst eine weitere Verschlüsselungsstufe bietet. Die Verschlüsselung kann sowohl vom Master- als auch von vom Slave-Endgerät beantragt werden, gestartet wird sie aber immer vom Master-Endgerät. Hierfür werden zwischen dem Master und dem Slave notwendige Parameter, wie z.B. die Schlüssellänge, vereinbart und daraus ein Verbindungsschlüssel generiert. Dieser Verbindungsschlüssel ist dann, neben einem *Cipher Offset* und einer Zufallszahl, Teil des Schlüssels mit dem die übertragenen Daten chiffriert werden. Als *Cipher Offset* dient hier beispielsweise die *Bluetooth Device Address* des Masters. Letztendlich wird ein Stromchiffre eingesetzt, der im *Bluetooth*-Standard als *E0* bezeichnet wird. Darüber hinaus wird für jedes übermittelte Paket ein eigener Schlüssel aus der *Bluetooth Device Address* und dem Zeittakt des Masters berechnet. Die Verschlüsselung der Daten findet allerdings nur während der Funkübertragung statt. Auf dem Weg von der Applikation zur Funkschnittstelle auf Seiten des Senders und von der Funkschnittstelle zur Applikation des Empfängers sind die Daten komplett unverschlüsselt. Allgemein erlaubt der *Bluetooth*-Standard drei Modi für die Sicherheit der *Bluetooth*-Endgeräte. Im ersten Modus sind gar keine Sicherheitsfunktionen beim Endgerät eingestellt, es reagiert aber auf Authentisierungsanfragen anderer Endgeräte. Im zweiten Modus ist es von den Einstellungen für die Applikationen und die Vertrauenswürdigkeit — *trusted* oder *untrusted* — eines anfragenden *Bluetooth*-Endgerätes abhängig, ob Sicherheitsmechanismen in Gang gesetzt werden oder nicht. Das Endgerät reagiert auch in diesem Modus auf Authentisierungsanfragen anderer Endgeräte. Im dritten Modus sind die Sicherheitsmechanismen generell eingeschaltet und das Endgerät verlangt eine gesicherte Verbindung zu anderen *Bluetooth*-Endgeräten. Generell ist es noch möglich zu definieren, ob das Endgerät überhaupt von anderen *Bluetooth*-Endgeräten gefunden wird, oder ob es unsichtbar bleibt. Man unterscheidet hier zwischen „immer und für alle sichtbar“, „nur für ausgewählte Endgeräte sichtbar“ und „unsichtbar“.

Trotz dieser Sicherheitsfunktionen birgt die *Bluetooth*-Technologie einige Schwächen im Sicherheitskonzept. So ist eine Verschlüsselung nicht grundlegend vorgeschrieben und auch die Hersteller haben diesen Punkt als Voreinstellung meist deaktiviert. Darüberhinaus sind auch die voreingestellten PINs, sehr oft „0000“, in den Handbüchern der Endgeräte nachzulesen, die mitunter auch im Internet veröffentlicht werden. Bei Endgeräte ohne Eingabemöglichkeit, wie z.B. Druckern oder Headsets, ist es sogar unmöglich diese unsichere PIN zu ändern. Es ist keine Standardlänge für die PIN definiert und auch die Komplexität der PIN wird nicht nachgeprüft. Einfache PINs, wie „1234“ oder eben „0000“, sind somit leicht zu erraten und ein Angreifer kann ohne Probleme über eine *Man-in-the-Middle*-Attacke die bestehende Kommunikation abhören. Durch gezielt eingesetzte Störsender kann die Kommunikation unmöglich gemacht oder mittels einer *Denial-of-Service*-Attacke das *Bluetooth*-Endgerät sogar total zum Ausfall gezwungen werden. Diese als *Bluesnarfing*[4] oder *Bluejacking*[5] genannten Techniken sind derzeit gängige Verfahren, um die *Bluetooth*-Kommunikation zu beeinträchtigen. Auch die Mechanismen des Zufallsgenerators sind im *Bluetooth*-Standard nicht klar definiert und variiert stark zwischen Endgeräte unterschiedlicher Hersteller. [7]

Es ist also noch viel an der Sicherheit von *Bluetooth* zu verbessern, was in der Weiterentwicklung des Standards bereits berücksichtig wird. Auch die Hersteller müssen mehr dazu angehalten werden, bereits bestehende Sicherheitsfunktionen standardmaßig zu aktivieren, um zumindest einen Grundschutz für das *Bluetooth*-Endgerät zu gewährleisten.

[4] engl. to snarf=*klauen, stehlen*
[5] von engl. hijacking=*Entführung*

4.2.3 Wireless LAN

Eine weitere kabellose Verbindung stellt das *Wireless LAN*, kurz *WLAN* dar. *WLAN* wurde erstmals 1997 von dem *Institute of Electrical and Electronic Engineers* (*IEEE*) als *IEEE 802.11* definiert. Ein *WLAN* entspricht in der Netzwerktechnik einem Ethernet mit TCP/IP-Kommunikation, mit dem einzigen Unterschied, dass die Endgeräte nicht kabelgebunden, sondern per Funk über *Access-Points* an das Netzwerk angebunden sind. Die Installation eines *WLAN*s gestaltet sich sehr einfach, da eine aufwendige Verkabelung meist vollständig entfällt. In vielen mobilen Endgeräten, vornehmlich Laptops, aber auch PDAs und seit Kurzem auch in Mobiltelefonen und Smartphones, ist die Unterstützung für die Version *IEEE 802.11b* des *WLAN*-Standards eingebaut. In aktuelleren Endgeräte wird aber verstärkt die stark überarbeitete Version *IEEE 802.11g* eingesetzt. Die Hersteller von *WLAN*-Systemen haben sich in der *Wi-Fi*-Allianz zusammengeschlossen und vergeben für *WLAN*-kompatible Endgeräte das *WiFi*-Zertifikat.

Eine Funkverbindung über *WLAN* kann über zwei Verfahren realisiert werden. Das erste Verfahren is der „Ad-hoc-Modus“, bei dem zwei oder mehr Clients direkt miteinander kommunizieren können. Dies kann beispielsweise bei der Kommunikation zwischen PDA und Laptop zur Datensynchronisation eingesetzt werden. Das zweite Verfahren ist der „Infrastructure-Modus“, der am häufigsten eingesetzt wird. Hier erfolgt die Kommunikation über einen zentralen Zugangspunkt, dem *AccessPoint*. Dabei kann der *AccessPoint* als Verteiler für mehrere Clients dienen, aber auch als Zugangs-Router zu einem kabelgebundenen Netzwerk. Im Endanwenderbereich ist in einem *AccessPoint* oft auch ein Modem eingebaut, mit dem z.B. der Zugang ins Internet ermöglicht wird. Um eine breitere Netzabdeckung, z.B. auf einem Firmengelände, zu erreichen, kann mit mehreren *AccessPoints* Funkzellen errichtet werden, so dass überall der Zugang zum Firmennetzwerk möglich ist. Die Reichweite der Funkzelle ist aber, wie allgemein bei *WLANs*, stark von den Umgebungsbedinungen abhängig und kann zwischen 10 und 300 m schwanken. In Gebäude eingebaute Stahlträger oder Brandschutztüren können unüberwindbare Hindernisse darstellen. Man kann die *AccessPoints* aber auch als *Repeater* einsetzen, um die Reichweite einer Funkzelle zu erhöhen. Wenn eine größere Distanz überbrückt werden muss, dann können *AccessPoints* auch mit speziellen Komponenten, wie z.B. Richtfunkantennen, ausgestattet werden und erreichen somit Reichweiten im Kilometerbereich. Das ist aber je nach Hersteller und *AccessPoint* verschieden. *WLAN* funkt dabei, genau wie *Bluetooth* im 2 GHz Frequenzband mit 13 Kanälen in einem Abstand von 5MHz und hat eine Sendeleistung von maximal 100 mW. Im Standard *IEEE 802.11b* wird eine maximale Übertragungsrate von 11 MBit/s, beim Standard *IEEE 802.11g* eine maximale Übertragungsrate von 54 MBit/s erreicht. Diese Werte sind jedoch auch wieder stark von der jeweiligen Umgebung abhängig und auch von der Distanz zum *AccessPoint*.

Im allgemeinen Standard *IEEE 802.11* sind diverse Sicherheitsmechanismen definiert, die durch die Erweiterungen *b* und *g* nicht verändert oder ergänzt werden. Erster Mechanismus ist die Vergabe eines Netzwerknamens, der *Service Set Identifier* (*SSID*). Die *SSID* kann dabei fest vorgegeben werden, wobei nur Zugriffe von Endgeräte mit gleicher *SSID* gestattet werden, oder in einem Automatik-Modus jede *SSID* akzeptiert werden. Von den Herstellern ist meist in den *AccessPoints* eine feste *SSID*, wie z.B. „default“ oder „WLAN“,

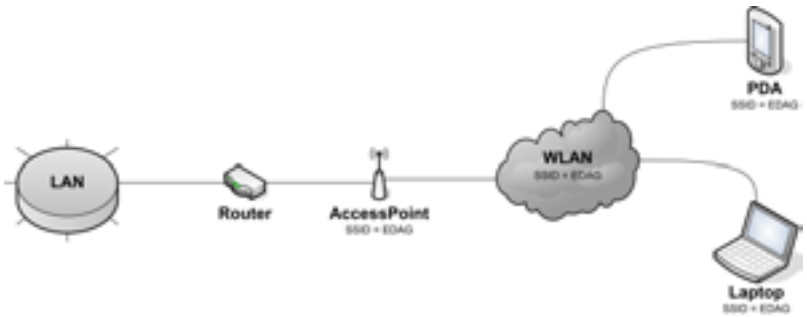


Abb. 4.2: *Beispielhafter Aufbau eines WLANs*

eingestellt. Wenn man die *AccessPoints* als Funkzelle betreibt, dient die *SSID* auch dazu, die nächste Funkzelle zu finden. Laut *IEEE 802.11* kann auch der Broadcast, also das automatische Senden der *SSID*, unterdrückt werden, was aber standardmäßig von den Herstellern nicht aktiviert wird. Zweiter Mechanismus ist der Zugriffsschutz über MAC-Adressen. Dabei kann eine Liste mit festen MAC-Adressen definiert werden, die Zugang auf den *AccessPoint* und damit auf das *WLAN* haben. Eine Pflege der Adresslisten muss dabei manuell für jeden einzelnen *AccessPoint* erfolgen, was in größeren Netzen mit vielen Funkzellen oder vielen Teilnehmern meist nicht durchfürbar ist. Das Filtern nach MAC-Adressen ist aber standardmäßig nicht vorgesehen und wird von einigen Herstellern nicht unterstützt.

Wichtiger ist die Verschlüsselung der Datenkommunikation, sowie die Überprüfung der Integrität, aber auch die Authentifizierung gegenüber dem *WLAN*. In den Spezifikationen von *IEEE 802.11* ist hierfür das *Wired Equivalent Privacy*-Protokoll (*WEP*) definiert. Als Basis für die Verschlüsselung der übertragenen Daten verwendet *WEP* den Stromchiffre *RC4*. Bei *RC4* wird dabei jedes Datenpaket mit einem Schlüssel und einem Initialisierungsvektor codiert. Der Schlüssel hat dabei eine Länge von 40, wahlweise aber auch 104 Bit. Dabei wird der Schlüssel alle kommunizierenden Partnern vom *AccessPoint* zur Verfügung gestellt bzw. fest in die Konfiguration des Teilnehmers eingetragen. Für die komplette Kommunikation innerhalb eines *WLAN*s wird dabei ein einziger gemeinsamer Schlüssel verwendet. Nur der Initialisierungsvektor wird für jedes Datenpaket neu berechnet und hat bei *WEP* eine Länge von 24 Bit. Zur Verschlüsselung des Datenpakets werden der Schlüssel und der Initialisierungsvektor mit einander verkettet und daraus eine zufällige Zahl als Bitstrom generiert. Der Bitstrom hat, je nach Länge des Schlüssels, eine Länge von 64 bzw. 128 Bit. Für eine spätere Integritätsprüfung des Datenpakets wird davon eine 32 Bit lange CRC-Prüfsumme erstellt und mit dem Datenpaket verkettet. Durch eine XOR-Verknüpfung aus dem Datenpaket mit der CRC-Prüfsumme und dem zufällig generierten Bitstrom wird nun das verschlüsselte Paket generiert. An dieses verschlüsselte Paket wird der Initialisierungsvektor unverschüsselt vorangestellt und das Paket übertragen. Der Empfänger benutzt nun diesen Initialisierungsvektor um seinerseits einen Bitstrom zu generieren und macht eine XOR-Verknüpfung mit den codierten Daten und erhält somit wieder das ursprüngliche Datenpaket. Da die Daten frei in die Luft übertragen werden muss nun überprüft werden, ob ein anderer Teilnehmer authorisiert ist, die Daten zu empfangen. Dafür sind in *WEP* zwei Authentifizierungsmodi definiert. Zum einen gibt es den „Open"-Modus, bei dem keine Authentifizierung statt findet. D.h. jeder Teilnehmer, der das *WLAN* empfängt, darf sich auch damit verbinden. Der zweite Modus ist der „Shared Key"-Modus, bei dem der *AccessPoint*

ein „Challenge-Response-Verfahren“ einsetzt. Dazu werden zufällig 128 Bytes erzeugt und an den Client geschickt. Der Client verschlüsselt diese Bytes mit dem bei ihm eingestellten Schlüssel, der auch zur Verschlüsselung der Datenpakete verwendet wird, und sendet sie zurück an den *AccessPoint*. Kann der *AccessPoint* die Bytes mit seinem eingestellten Schlüssel wieder entschlüsseln, so ist der Client authentisiert. Dabei muss sich immer nur der Client am *AccessPoint* authentisieren und nicht umgekehrt. Es handelt sich hier also um ein einseitiges Authentifizierungsverfahren.

Das *WEP*-Protokoll bietet nur einen sehr geringen Schutz und kann mit sehr einfachen Mitteln in sehr kurzer Zeit gebrochen werden. Vorallem der *RC4*-Algorithmus gilt schon seit 2001 als unsicher und sollte nicht mehr verwendet werden. Eine Attacke auf *WEP* kann dabei ohne großen Aufwand betrieben werden und es existieren diverse Tools, mit denen ein solcher Angriff realisiert werden kann. Durch den automatischen *SSID*-Broadcast des *WLAN*s erkennt der Angreifer zunächst, wo sich ein potentielles Opfer-Netzwerk befindet. Über das Tool *Netstumbler* ist diese Analyse z.B. unter *Microsoft Windows* möglich, ernsthafte Angreifer verwenden *Kismet* unter *Linux*. *Kismet* hat den Vorteil, dass es zusätzlich die empfangenen Datenpakete protokolliert und auf ihre Verschlüsselung testet. Denn nur verschlüsselte Pakete sind interessant, um den benötigten Schlüssel zu bekommen, damit man sich mit dem *WLAN* verbinden kann. Theoretisch sind ca. 5 Millionen Pakete notwendig, um eine sichere Attacke gegen das *RC4*-Verfahren zu fahren. Als Angreifer kann man aber selbst entscheiden, wann *Kismet* genügend verschlüsselte Pakete empfangen hat. Die Protokolldatei mit den gesammelten Paketen wird nun an das Tool *WEPAttack* übergeben. *WEPAttack* hat einmal die Möglichkeit mit einer *Wordlist*-Attacke den gesuchten Schlüssel zu finden. Dabei gibt es *Wordlists* mit bis zu 4 Milliarden Einträgen an möglichen Schlüsseln. Wird *WEPAttack* über die *Wordlist*-Methode nicht fündig, so hat man noch die Möglichkeit über statistische Verfahren die Zeichenkombination für den Schlüssel selbst zu generieren. Die Dauer der Entschlüsselung hängt ganz von der Komplexität des Schlüssels ab und kann von einer Sekunde bis hin zu mehreren Tagen gehen.

Hier ist schon die größte Schwachstelle eines *WLAN*s dargestellt. Die kurze Schlüssellänge von 40 Bit sichert ein *WLAN* nur sehr ungenügend und ein geeigneter Zugangsschlüssel ist in recht kurzer Zeit aus den protokollierten Daten zu extrahieren. Besser ist die Verwendung eines 104 Bit langen Schlüssels. Auch die Länge des Initialisierungsvektors von 24 Bit ist sehr kurz. Da der Initialisierungsvektor für jedes zu übermittelnde Paket neu berechnet wird, sind die Möglichkeiten von $2^{24} = ca.16,8Mio$ unterschiedlichen Initialisierungsvektoren bei ca. 4.000 Paketen ausgeschöpft und ab dann ist mit Wiederholungen zu rechnen. Da *WEP* diese Wiederholungen zulässt, kann ein Angreifer immer den gleichen Initialisierungsvektor zur Verschlüsselung der Daten verwenden. Die Berechnung des Initialisierungsvektors geht nämlich immer vom Sender aus und dadurch merken die anderen Teilnehmer nichts von dem Angriff. Der Angreifer muss dazu nur verschlüsselte Pakete abfangen, der Initialisierungsvektor befindet sich immer am Anfang des Pakets und hat eine konstante Länge von 24 Bit. Alternativ kann auch die Bitfolge aus Schlüssel und Initialisierungsvektor aus dem verschlüsselten Paket extrahiert werden. Mit dieser Bitfolge können vom Angreifer ebenfalls Datenpakete gefälscht werden. Dies ist zumindest bis zum nächsten Wechsel des Schlüssels möglich. Auch die Authentifizierung kann komplett gefälscht werden. Dazu muss vom Angreifer nur ein vollständiges Authentifizierungsprotokoll abgehört werden. Mit diesen Daten kann mit einer XOR-Verknüpfung aus *Challenge* und *Response* ein Bitstrom berechnet werden, der als Grundlage für weitere *Response*-Berechnungen zu einem gegebenen *Challenge* verwendet werden kann. Da sich ein *WLAN* unkontrolliert ausbreitet und die Funkwellen

z.B. keine Gebäudegrenzen kennen, sind alle diese Angriffe möglich, ohne dass der Angreifer sich direkt auf dem Firmengelände oder im Gebäude aufhält. Diese als *WarDriving* bezeichnete Methode ist derzeit gängige Praxis und stellt eine sehr hohe Bedrohung dar. Dadurch können z.B. wichtige lokale Daten ausspioniert werden oder weitere Angriffe auf das interne Netzwerk gestartet werden. Denn bei einer Standard-Konfiguration befindet sich der Angreifer direkt im internen Netzwerk sobald er im *WLAN* authentisiert ist.

Diese Schwachstellen sind bereits seit 2001 der *Wi-Fi*-Allianz bekannt, aber dennoch hat die Entwicklung eines Nachfolge-Standards für *WLAN*s mit verbesserten Sicherheitsfunktionen bis Mitte 2004 gedauert. Der daraus resultierenden Standard *IEEE 802.11i* verwendet *Wi-Fi Protected Access* (*WPA*) als Sicherheitsfunktion. Größte Unterschiede zwischen *WEP* und *WPA* ist die Verwaltung von dynamischen Schlüsseln durch das *Temporal Key Integrity Protocol* (*TKIP*) und eine bessere Benutzerauthentifizierung durch das *Extensible Authentication Protocol* (*EAP*). Bei *TKIP* wird ein *Zugangs-Schlüssel* nur zur Initialisierung der Verbindung benutzt. Für die eigentliche Datenübertragung wird ein *Session-Key* erzeugt, der nur den beiden Teilnehmern bekannt ist, die gerade die Daten austauschen. Der *Zugangs-Schlüssel* kann hier über einen zentralen Server verwaltet werden. Dabei kann für jeden Teilnehmer ein eigener *Zugangs-Schlüssel* verwaltet werden. Alternativ können auch sog. *Pre-Shared-Keys* allen Teilnehmern zur Verfügung gestellt. Da diese Methode auch in *WEP* verwendet wird, können bei dieser Einstellung auch Endgeräte Zugang zum *WLAN* erhalten, die *WPA* noch nicht unterstützen.

Bei *EAP* werden zunächst drei Komponenten unterschieden. Erste Komponente ist der *Supplicant*, der einen Zugang zu einem Netzwerk anfragt. Zweite Komponente ist der *Authenticator*, der die Identität des *Supplicants* überprüft. Dritte Komponente ist der *Authentication Server* der die eigentliche Zugangsberechtigung des *Supplicant* überprüft. In einem *WLAN* ist der *Supplicant* das mobile Endgerät, also Laptop, PDA usw., der *Authenticator* ist der *Access Point* und als *Authentication Server* kann z.B. ein *RADIUS*-Server eingesetzt werden. Die Authentifizierung ist auf einzelne Ports des *Authentication Servers* bezogen, für die eine zweigeteilte Kontrolleinheit, die *Port Access Entity*, existiert. Die beiden Teile der *Port Access Entity* bestehen aus einem *Controlled Port* und einem *Uncontrolled Port*. Über den *Uncontrolled Port* hat der *Supplicant* die Möglichkeit sich am *Authenticator* zu identifizieren, einen Zugriff zum Netzwerk erhält er darüber jedoch nicht. Der Zugang erfolgt ausschließlich über den *Controlled Port*, der bis zur vollständigen Authentifizierung des *Supplicants* geschlossen bleibt. *EAP* kennt unterschiedliche Methoden, um die Authentifizierung selbst zu verschlüsseln. Eine Methode ist *EAP-MD5*, bei der der *Authentication Server* eine zufällig generierte Zahl an den *Supplicant* schickt, die dieser mit einem gemeinsam bekannten geheimen Schlüssel nach dem *MD5*-Algorithmus codiert und an den *Application Server* zurück schickt. In einem *WLAN* ist aber die zweite Methode, *EAP-TLS*, besser geeignet. Dabei findet ein Zertifikatsaustausch zwischen *Supplicant* und *Application Server* statt. Hierfür ist eine *Public Key Infrastructure* Voraussetzung. Der danach ausgehandelte *Session-Key* ist über *TLS*[6] geschützt, was auch ein abhören der übertragenen Daten erschwert. Alle Spezifikationen zu *EAP* sind in *RFC 2284* definiert und können unter [1] nachgelesen werden.

Aktuelle *WLAN*-Endgeräte unterstützen diesen neuen Standard allerdings noch nicht. Am weitesten verbreitet sind derzeit *IEEE 802.11b* und *IEEE 802.11g*. Wenn ein *WLAN* in einem Unternehmen eingesetzt werden soll, ist auf jeden Fall *IEEE 802.11g* oder bes-

[6] Mehr Informationen über *TLS*: *siehe* [9]

Abb. 4.3: *Aufbau von EAP*

ser *IEEE 802.11i* zu verwenden. Ebenso ist eine höchstmögliche Verschlüsselung zu aktivieren. Dabei sollte darauf geachtet werden, dass evt. auch ältere Endgeräte das *WLAN* nutzen können. Wird ein *WLAN* ungeschützt installiert, so ist ein Eindringen nämlich nicht zwangsläufig strafbar. Erst ein Eindringen in ein gesichertes *WLAN*, egal in welcher Art es geschützt ist, ist illegal. In Betracht kommen hier beispielsweise die Paragraphen §265a StGB („Erschleichen von Leistungen"), §202a StGB („Ausspähen von Daten"), §303a StGB („Datenveränderung") und §303b StGB („Computersabotage"). Bei den momentanen Möglichkeiten ein *WLAN* anzugreifen sollten keine sicherheitsrelevanten Daten über ein *WLAN* erreichbar sein. Lediglich der Zugriff auf Internet und evt. Home-Verzeichnisse sind sinnvolle Möglichkeiten für den Einsatz eines *WLAN*s in einem Unternehmen. Das restliche Netzwerk sollte nochmal extra abgesichert werden. Der Einsatz von einem *WLAN* sollte daher gut überlegt und die Notwendigkeit genau analysiert werden.

4.2.4 Mobilfunk

Der Zugang über Mobilfunk ist natürlich nur den Mobiltelefonen und den Smartphones vorbehalten. Aber auch für Laptops gibt es Erweiterungskarten, um Daten über das Mobilfunknetz zu versenden. Dafür stehen in Deutschland derzeit zwei unterschiedliche Netze zur Verfügung, das *GSM* und das neue *UMTS*. *GSM* ist bereits seit 1992 in Deutschland aktiv und ist mit einer Netzabdeckung von weltweit ca. 80 Prozent der meistverbreitete Standard. Das *GSM*-Netz funkt in den drei unterschiedlichen Frequenzen 900 MHz (weltweit), 1.800 MHz (ebenfalls weltweit) und 1.900 MHz (nur in Amerika). Für die Datenübertragung steht dabei eine Bitrate von maximal 9,6 kbit/s zur Verfügung. Da darüber eine Datenübertragung nur sehr langsam abläuft, wurden speziell für die Datenübertragung die Erweiterungen *GPRS* und *HSCSD* entwickelt. Beim paketorientierten *GPRS* erreicht man durch die Bündelung von acht *GSM*-Kanälen eine Bitrate von maximal 171,2 kbit/s. Das leitungsorientierte *HSCSD* kommt ebenfalls durch die Bündelung von acht *GSM*-Kanälen auf maximal 115,2 kbit/s. Von aktuellen Mobiltelefonen wird hauptsächlich *GPRS* unterstützt.

Eine noch schnellere Datenübertragung bietet aber das neue Mobilfunknetz *UMTS*, das seit Anfang 2004 auch in Deutschland verfügbar ist. Erst zum Ende dieses Jahres werden die ersten Mobiltelefone und *PCMCIA*-Karten für Laptops von den Mobilfunkbetreibern angeboten. Bei *UMTS* kann eine maximale Datenübertragungsrate von 384 kbit/s erreicht werden. Während meiner Tätigkeit bei der *EDAG Engineering & Design AG* hatte ich im August 2004 die Möglichkeit einer dieser neuen *UMTS-PCMCIA*-Erweiterungskarten für Laptops zu tes-

Abb. 4.4: *Vodafone Mobile Connect Card UMTS*

ten, die *Vodafone Mobile Connect Card UMTS*. Der komplette Testbericht ist unter [25] nachzulesen und im Anhang dieser Diplomarbeit zu finden. Aufgrund dieses ausführlichen Tests wird an dieser Stelle darauf verzichtet das Thema Mobilfunk näher zu beschreiben.

4.3 Remote Access

Als *Remote Access* wird allgemein der externe Zugang zu einem Netzwerk bezeichnet. Dabei hat ein Mitarbeiter die Möglichkeit sich beispielsweise von zu Hause mit dem Unternehmensnetzwerk zu verbinden und kann dann alle Möglichkeiten nutzen, die er auch im Büro hat. Ein Unternehmen kann einen *Remote Access* durch zwei Möglichkeiten anbieten. Zum einen können lokale Einwahlknoten zur Verfügung gestellt werden. Dazu muss der Mitarbeiter eine vorgegebene Telefonnummer wählen und ist dann mit seinem Modem direkt mit dem Firmennetzwerk verbunden. Als Zugangstechnik dienen hier ein analoger Telefonanschluß mit einer Datenrate von bis zu 56 kbit/s oder ein ISDN-Anschluß der durch Kanalbündelung bis zu 128 kbit/s an Datenrate erreicht. Dieser Zugang ist vergleichbar mit einem *Internet Service Provider* bei dem ebenfalls eine spezielle Einwahlnummer erforderlich ist, um sich mit dem Internet zu verbinden. Bei der Einwahl ins Unternehmensnetzwerk ist man direkt damit verbunden und es gelten alle Regeln, die auch gelten würden, wenn man an seinem Arbeitsplatz im Unternehmen sitzen würde. Dies ist natürlich ein sehr sicherer Weg ein *Remote Access* anzubieten. Die Sicherheitsmaßnahmen, wie Spam-Filter, Virenschutz und Firewall, greifen sofort auch für alle über *Remote Access* angebundene mobilen Endgeräte und man läuft weniger Gefahr, dass sich Mitarbeiter ungewollte Viren, Würmer und Trojaner einfangen. Jedoch ist diese Variante auch verwaltung- und kostenintensiver. Es muss rund um die Uhr möglich sein, sich mit dem Unternehmensnetz zu verbinden und meist genügt auch nicht die Bereitstellung von nur einem Zugangs-Port. All das bedeutet einen hohen Aufwand um ein *Remote Access* zu betreiben.

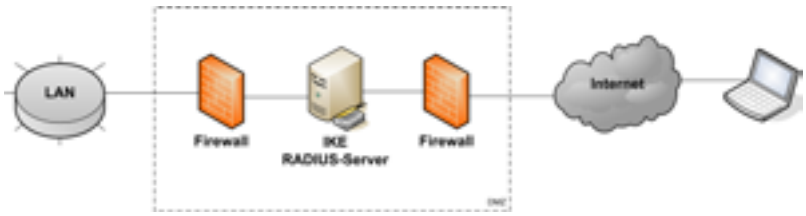


Abb. 4.5: *Remote Access-Lösung am Beispiel von IPSec*

Zweite Möglichkeit ist eine Verbindung direkt über das Internet aufzubauen. Dabei wählt sich ein Mitarbeiter bei dem *Internet Service Provider* seiner Wahl ein und verbindet sich dann mit dem Unternehmensnetzwerk. Der Vorteil an dieser Variante ist, dass auch schnellere Zugangstechniken wie z.B. *DSL* genutzt werden können. Natürlich muss diese Verbindung über einen gesicherten Weg statt finden, da ansonsten das Unternehmensnetzwerk mit allen sensitiven Daten dem ganzen Internet zur Verfügung steht. Diese *Remote Access*-Verbindung kann daher über den Aufbau eines *Virtual Private Network* (*VPN*)-Tunnels gesichert werden. Für den Aufbau eines *VPN* sind derzeit zwei Protokolle im praktischen Einsatz. Das erste Protokoll ist das von *Microsoft* entwickelte *Point-to-Point Tunneling Protocol* (*PPTP*). *PPTP* setzt dabei direkt auf eine *PPP*-Verbindung auf, das für einen Verbindungsaufbau

über Wählleitungen verwendet wird. Beim Zugang über *DSL* wird die Variante *PPPoE* verwendet. In der *PPTP*-Architektur werden zwei wesentliche Komponenten unterschieden. Einmal gibt es den *PPTP Access Concentrator*, der normalerweise direkt im Client implementiert ist. Auf der anderen Seite gibt es einen *PPTP Network Server*, der alle Anfragen des *PPTP Access Concentrator* verwaltet und verteilt. Damit ein Client über *PPTP* eine *VPN*-Verbindung aufbauen kann, muss er sich zunächst beim *PPTP Network Server* authentifizieren und bekommt bei Erfolg eine IP-Adresse aus dem Unternehmensnetzwerk zugewiesen. Nun ist der Client Teil des Unternehmensnetzwerkes und kann Daten durch den *VPN*-Tunnel senden und empfangen. Bei *PPTP* werden die Daten aber nicht verschlüsselt übertragen. Zwar kann eine gesicherte Verbindung über *SSL* oder *TLS* aufgebaut werden, aber das hat nichts mit *PPTP* zu tun. Sollte eine direkte Verschlüsselung der Daten statt finden, bevor diese durch den *VPN*-Tunnel geschickt werden, so ist das bereits beim Aufbau der *PPP*-Verbindung zu definieren. Hier steht aber bestenfalls eine Verschlüsselung nach *RC4* zur Verfügung, was allgemein als unsicher gilt. Erste Schwachstellen von *PPTP* urden bereits 1998, zwei Jahre nach Veröffentlichung des Protokolls, vor allem im Authentifizierungsverfahren, festgestellt [28]. Dennoch wird *PPTP* weiterhin als *VPN*-Protokoll eingesetzt, was aber nicht zu empfehlen ist. Alle technischen Informationen zu *PPTP* wurden unter der informellen *RFC 2637* veröffentlicht und sind unter [13] nachzulesen.

Besser eignet sich *IPSec* für den Aufbau eines *VPN*-Tunnels. *IPSec* ist eigentlich dazu entworfen worden, um Sicherheitsschwächen im *IP*-Protokoll zu beheben. Dabei stellt *IPSec* die Vertraulichkeit, Authentizität und Integrität der übermittelten Daten sicher. *IPSec* zeichnet sich vorallem darin aus, dass die Authentifizierung über einen Schlüsselaustausch realisiert wird. Dafür zuständig ist der *Internet Key Exchange*, kurz *IKE*. Als *IKE*-Stelle wird meist ein Authentifizierungsserver, wie z.B. einen *RADIUS*-Server, eingesetzt. *IKE* übernimmt dabei die automatische Schlüsselverwaltung, und einigt sich mit dem anfragenden Client darauf, welcher Verschlüsselungsalgorithmus zur Authentifizierung und zur späteren Datenübertragung verwendet werden soll. Dabei arbeitet *IPSec* mit unterschiedlichen symmetrischen und asymmetrischen Verschlüsselungsverfahren. Der eigentliche Schlüsselaustausch findet über das *Diffie-Hellman Key Agreement* statt. Das *Diffie-Hellman Key Agreement* eignet sich besonders für den Schlüsselaustausch bei unsicheren Netzen, wie z.B. dem Internet. Dabei einigen sich die Kommunikationspartner auf eine Primzahl p und eine weitere Zahl mit der Eigenschaft $g \mod p$, wobei gilt $2 \leq g \leq p-2$. Diese beiden Zahlen müssen nicht geheim sein und können über das unsichere Internet übertragen werden. Die beiden Teilnehmer generieren nun eine geheime Zufallszahl a bzw. b, wobei a und b aus der Menge $\{0, \ldots, p-2\}$ stammen müssen. Nun werden die Zahlen $A = g^a \mod p$ und $B = g^b \mod p$ berechnet und zum jeweiligen Gegenüber übertragen. Beide Kommunikationspartner berechnen nun $B^a = (g^b)^a = (g^a)^b = A^b = K$, wobei K auf beiden Seiten gleich ist und als Kommunikationsschlüssel benutzt wird[7].

In *IPSec* wird das *Diffie-Hallman Key Agreement* zur Aushandlung einer *Security Association* genutzt. Diese Aushandlung findet in zwei Phasen statt, wobei für die erste Phase zwei Modi und für die zweite Phase ein Modus zur Verfügung stehen. Die erste Phase dient ausschließlich dazu die Authentifizierungsmodalitäten zu vereinbaren. Dazu schickt im *Main Mode* der Anfragende einen oder mehrere Vorschläge für Authentifizierungs- und Verschlüsselungsalgorithmen an den *IKE*-Server. Diese wählt nun einen Vorschlag aus und bestätigt diesen dem Anfragenden. Anfragender und *IKE*-Server generieren nun einen Zufallswert,

[7] Eine komplette technische Beschreibung des *Diffie-Hallman Key Agreement* ist unter [22] nachzulesen

den sog. *Nonce*, und senden diesen Wert mit dem öffentlichen Teil des *Diffie-Hallman Key Agreement* an sein Gegenüber. Der Anfragende berechnet die *Diffie-Hallman*-Signatur und schickt sie zusammen mit seiner Identität symmetrisch verschlüsselt an den *IKE*-Server. Dieser schickt ebenfalls seine Daten symmetrisch verschlüsselt an den Anfragenden. In der zweiten Phase wird nun im *Quick Mode* die Identität der beiden Komponenten überprüft und eine *Security Association* erstellt. Der Datenaustausch erfolgt in dieser zweiten Phase ausschließlich verschlüsselt. Alternativ steht in der ersten Phase noch der *Aggressive Mode* zur Verfügung, bei dem der Austausch der Identitäten und des *Nonce* unverschlüsselt erfolgt und damit nicht zu empfehlen ist. Zweiter Sicherheitsaspekt in *IPSec* ist der *Authentication Header*. Er soll die Integrität der Daten sicher stellen und erschwert einen *Replay*-Angriff. Bei diesem *Replay*-Angriff wird einfach ein bereits aufgezeichneter Datenstrom noch einmal gesendet und somit versucht, sich als Kommunikationspartner auszugeben. Um sich davor zu schützen wird im *Authentication Header* ein Feld mit der jeweiligen Sequenznummer geführt, das in aufsteigender Folge vom Absender des Pakets vergeben wird. Taucht plötzlich ein Paket mit dem gleichen *Security Parameters Index* auf, so wird dieses Paket verworfen. Der *Security Parameters Index* identifiziert die vorher vereinbarte *Security Association*. Ähnliche Ziele verfolgt der *Encapsulated Security Payload*. Auch hier schützt eine Sequenznummer einen *Replay*-Angriff und das Paket wird über einen *Security Parameters Index* identifiziert. Im Gegensatz zum *Authentication Header* werden hier aber die Informationen im eigentlichen *IP-Header* nicht in Betracht gezogen, sondern das komplette IP-Paket in den *Payload Data* übertragen. Der *Authentication Header* versucht hingegen auch alle möglichen Felder des *IP-Headers* zu schützen.

Die Spezifikationen zu *IPSec* sind sehr umfangreich und umfassen mehrere *RFCs*. In *RFC 2401* [15] werden alle Grundlagen definiert und beispielsweise in *RFC 2409* [14] wird der *IKE* beschrieben. *IPSec* ist ein sehr sicheres Sicherheitsprotokoll für eine *VPN*-Verbindung, wobei die komplizierte Architektur des *IKE* oftmals bemängelt wird. *IPSec* ist jedoch, im Gegensatz zu *PPTP*, bereits für das neue *IPv6* gerüstet, das langsam eingeführt wird. Auch *Microsoft* bietet in seinen Betriebssystemen standardmäßig die Unterstützung für *IPSec* und empfiehlt dessen Verwendung. Allgemein ist grundsätzlich *IPSec* für eine sichere *VPN*-Verbindung zu empfehlen.

Abb. 4.6: *Beispielhafter Aufbau einer VPN over WLAN-Lösung*

Eine *VPN*-Verbindung hat allgemein den Vorteil, dass es während einer TCP/IP-Verbindung eine sichere Kommunikation ermöglicht. So ist ein *VPN*-Tunnel auch während einer *WLAN*-Verbindung möglich und bietet dadurch eine weitere Sicherheitsstufe beim Einsatz dieser Technologie. Voraussetzung für *VPN over WLAN* ist, dass sich die beiden *VPN*-Gegenstellen, bei *IPSec* z.B. der Client und die *IKE*-Gegenstelle, im gleichen Subnetz befinden und somit ein Verbindungsaufbau möglich ist. Die Kommunikation geht dann vom *Wireless Client* über den *AccessPoint* des *WLAN*s zu der im Netz befindlichen Gegenstelle. Diese Gegenstelle kann nun entweder kabelgebunden oder ebenfalls kabellos am Netzwerk angeschlossen sein. In großen Firmennetzwerken stellt meist ein *VPN*-Server die zentrale

Gegenstelle dar, der den Zugang zum Netzwerk ermöglicht. Mittels Routern wird dann die komplette *VPN over WLAN*-Kommunikation mit dem Netzwerk über diesen *VPN*-Server durchgeführt. Verbindet man den *VPN*-Server direkt mit dem *AccessPoint*, sichert mna somit jede TCP/IP-Kommunikation über *WLAN* durch dieses Verfahren. So kann *WLAN* auch sicher in Unternehmen eingesetzt werden, das sicherheitsrelevante Informationen verarbeitet.

Kapitel 5

Software-Produkte für mobile Endgeräte

Wenn mobile Endgeräte in einem Unternehmen einsetzen werden sollen, so ist das kein Problem, solange es in einem überschaubaren Rahmen getan wird. Man kann jedes Endgerät einzeln administrieren, es bei dem Benutzer einrichten und auch bei Fragen behilflich sein. Übersteigt es aber eine gewisse Zahl, so sind diese Leistungen nicht mehr ohne einen gewissen Aufwand möglich. Schnell hat man es dann mit unterschiedlichen Endgeräten zu tun, die zum einen eine unterschiedliche Hardware-Ausstattung und zum anderen unterschiedliche Betriebssysteme haben.

Einfacher ist es für Unternehmen eine zentrale Verwaltung für möglichst viele Arten von mobilen Endgeräten zu finden, damit der Aufwand diese Endgeräte zu administrieren und zu betreuen deutlich geringer wird. Ein komplettes Konzept umfasst hierbei nicht nur die Möglichkeit einer zentralen Synchronisierung mit einer *Groupware*, sondern auch eine automatische Software-Verteilung für alle Systeme, eine zentrale Möglichkeit Einstellungen an den Geräten vorzunehmen, sowie automatisch Backups von den Endgeräten und den darauf enthaltenen Daten zu erstellen. Nicht zu vergessen ist auch ein komplettes Sicherheitskonzept, nicht nur für die Datenübertragung, sondern auch für die Daten, die auf den Geräten liegen und der Zugriff auf diese Daten. Ein solches Konzept stellt der folgende Teil meiner Diplomarbeit dar.

Zwar gibt es für die einzelnen Systeme ausgereifte Lösungen für die unterschiedlichen Bereiche, aber das würde den administrativen Aufwand in keinem Fall schmälern, er würde im Gegenteil noch viel größer. So gibt es für die Daten-Synchronisation für *Windows CE*-basierte *PDAs* die Software *ActiveSync*. Die gleiche Aufgabe erfüllt bei einem *Palm OS*-basierten *PDA* die Software *HotSync*. Diese Software kann jeweils auch zum Sichern der auf dem *PDA* enthaltenen Daten genutzt werden. Um die Sicherheit auf *PDAs* zu erhöhen, gibt es weitaus mehr an unterschiedlichen Software-Produkten die in Frage kämen. Aber eben wieder für jedes System eine eigene Software.

5.1 Kriterien für eine Auswahl

Da es so viele unterschiedliche mobile Systeme und Lösungen für diese Endgeräte auf dem Markt gibt, müssen einige Kriterien definiert werden, damit die entsprechende Lösung in Betracht gezogen werden kann. Dabei müssen allgemeine Kriterien, die für alle Lösungen gelten, und spezielle Kriterien, die nur für Lösungen mit speziellen Aufgaben, wie beispielsweise Sicherheitslösungen, gelten, unterschieden werden.

Das wichtigste allgemeine Kriterium ist sicherlich die einheitliche Administrierung aller mobilen Endgeräte. Darunter fallen sowohl Laptops, als auch *PDAs* und *Smartphones*. Es sollten zusätzlich so viele Betriebssysteme wie möglich unterstützt werden. Die Administrierung muss dabei einfach bleiben. Durch eine zentrale Lösung soll vermieden werden, dass diverse Einstellungen erst mühsam gesucht werden müssen. Unterstützung sollte hier ein gutes Handbuch oder eine gute Online-Hilfe bieten. Wichtig ist dennoch, dass so viele Einstellungen wie möglich an dem mobilen Endgerät voreingestellt werden können. Der Benutzer darf nach Verteilung der Lösung nur noch eingeschränkte Rechte im Setzen verschiedener Einstellungen haben. So darf der Benutzer beispielsweise nicht das Recht haben, eine ungesicherte Verbindung zur Daten-Synchronisation aufzubauen oder sicherheitsrelevante Daten unverschlüsselt auf dem mobilen Endgerät abzulegen. Es muss aber auch möglich sein, die Benutzer unterschiedlichen Gruppen zuzuordnen, da evt. ein Administrator oder Mitglied des Vorstandes mehr Rechte auf dem mobilen Endgerät haben soll, wie ein „normaler“ Benutzer. Die Einstellungen sollen aber nicht die Bedienbarkeit des mobilen Endgerätes beeinträchtigen. Auch die Handhabung der jeweiligen Software sollte für den Benutzer recht einfach sein und eine Einarbeitung möglichst ohne langwierige Schulung erfolgen. Eben diese Benutzer, die ein mobiles Endgerät besitzen und somit für die Nutzung der jeweiligen Lösung zugelassen werden müssen, sollten aus einem bestehenden Benutzerverzeichnis importierbar sein. Besser wäre noch eine komplette Kopplung an ein bestehendes Benutzerverzeichnis, wobei nur die in der Lösung vorgenommenen Einstellungen auch dort gespeichert werden. Alle anderen Informationen über den Benutzer, wie Anmeldenamen, evt. Passwort aber auch der eigentliche Name und weitere Kontaktdaten sollten in der bereits vorhandenen Benutzerverwaltung verbleiben. Dadurch wird eine Redundanz und Unterschiede bei den Benutzerdaten vermieden. Ein entsprechender Server, der für die Daten-Synchronisation und die Einhaltung der Sicherheitsfunktionen verantwortlich ist, sollte unkompliziert in die bestehende Netzstruktur integrierbar sein. Optional ist noch eine Kombination von verschiedenen Lösungen sicherlich sinnvoll. Es ist z.B. zu prüfen, ob in eine Lösung für Daten-Synchronisation, Backup und Software-Verteilung evt. auch eine Lösung für Sicherheit integrierbar ist. Somit kann sicher gestellt werden, dass neue Sicherheitseinstellungen durch die Software-Verteilung auf den jeweiligen Endgeräten installiert werden.

Für eine Lösung, die für Daten-Synchronisation, Backup und Software-Verteilung zuständig ist, sollte die Möglichkeit bestehen direkt mit einem Groupware-Server zu kommunizieren. Auf einem Groupware-Server werden allgemein alle *PIM*-Daten zentral gespeichert, auf die man über einen Client zugreifen kann. Es gibt derzeit zwei weit verbreitete Groupware-Systeme, einmal *Microsoft Exchange* und *Lotus Notes*. Bei der *EDAG Engineering & Design AG* wird als Groupware *Lotus Notes* eingesetzt und es somit muss die Lösung mit dieser Groupware kommunizieren können. Voraussetzung ist ebenfalls eine gesicherte Verbindung während der Kommunikation zwischen Synchronisations-Server und Client. Die Verbindung muss dabei mindestens durch *SSL* oder *TLS* geschützt sein. Besser ist zusätzlich noch eine

VPN-Verbindung über einen *IPSec*-Tunnel. Hat die Lösung noch einen einheitlichen *Desktop Connector*, so ist das wiederum ein großer Vorteil. Somit entfallen die Installation von *ActiveSync* und *HotSync* auf einem Host-System, die dem Benutzer evt. Freiheiten bieten, die nicht gewollt sind. Optional ist auch eine Inventarisierung der mobilen Endgeräte und eine genaue Zuweisung zu einem bestimmten Benutzer wünschenswert. Dadurch können Einstellungen benutzerbezogen definiert werden, die sich auch auswirken, wenn das Endgerät den Besitzer wechselt.

Speziell für eine Sicherheitslösung gilt vor allem, dass aktuelle Sicherheitsstandards ausgewählt werden können. Bereits gebrochene Standards, wie z.B. *MD5* für die Integritätsprüfung oder *DES* für die asymmetrische Verschlüsselung, sollten dabei nicht verwendet werden. Als aktueller Sicherheitsstandard sind beispielsweise für die Überprüfung der Integrität *SHA* oder für die symmetrische Verschlüsselung *AES* anzusehen. Optional sollte für die Authentifizierung an dem Gerät oder für einen Verbindungsaufbau ein *Zwei-Faktor-Verfahren* möglich sein, ein einfacher Passwortschutz ist aber auch ausreichend. Ebenso optional ist die Einstellmöglichkeit bis auf Hardware- bzw. Schnittstellenebene. Wenn man über die Software die eingebaute Kamera oder die Unterstützung von *WLAN* oder *Bluetooth* explizit ausschalten kann, so steigert das durchaus die Sicherheit des mobilen Endgerätes.

Natürlich sind auch die Kosten ein wichtiger Punkt für die Auswahl einer geeigneten Lösung. In Betracht kommen hier zunächst einmal die eigentlichen Anschaffungskosten. Hier ist auch das verwendte Lizenzmodell relevant, das für die Ermittlung der laufenden Kosten wichtig ist. Die Lizenzmodelle beziehen sich hierbei entweder auf ein einzelnes Endgeräte, auch *Gerätelizenz* genannt, oder auf einen Benutzer, auch *Benutzerlizenz* genannt. Bei einer *Gerätelizenz* muss für jedes einzelne Endgerät auf dem die Lösung installiert wird Lizenzgebühren bezahlt werden. Bei einer *Benutzerlizenz* wird für jeden Benutzer bezahlt, der die Lösung benutzt, egal wie viele Geräte der Benutzer dazu verwendet. Ein *Benutzerlizenzmodell* ist hier zu bevorzugen, weil bereits dadurch meist Kosten eingespart werden können. Weiterhin ist zu überprüfen welche Service- und Support-Leistungen in dem jeweiligen Lizenzmodell, das meist mit einem Supportvertrag gleichzusetzen ist, enthalten sind und ob evt. Mehrkosten allein durch Supportanfragen entstehen. Auch die personellen Kosten sind ein wichtiger Punkt. Die Installation und Wartung der Lösungen sollte schnell und einfach gehen, damit wenig Aufwand von externen Technikern betrieben werden muss. Dieser Aufwand wird in der Regel als Manntag angegeben, wobei das für den Arbeitsaufwand eines Technikers pro Tag steht. Je mehr von internen Mitarbeitern geleistet werden kann, desto kalkulierbarer sind die Kosten[1].

Zum Schluß nochmal eine kurze Übersicht aller definierter Kriterien, sortiert nach ihrer Wichtigkeit:

1. Allgemeine Kriterien
   (a) einheitliche Administrierung
   (b) Administrierung möglichst aller mobilen Systeme
   (c) genügend Einstellungen für Benutzer vordefinierbar

[1] Alle angegebenen Preise sind Listenpreise und können sich durch eine Mengenstaffelung oder durch spezielle Angebote noch ändern. Außerdem verstehen sich alle Preise zzgl. der gesetzlichen MwSt. von derzeit 16%

   (d) einfache Handhabung für Administratoren und Benutzer
   (e) Kopplung mit bereits vorhandener Benutzerverwaltung
   (f) Integrierbarkeit in momentane Netzstruktur
   (g) Kombinierbarkeit aller Produkte (optional)

2. Kriterien für Synchronisation und Backup
   (a) Synchronisation mit *Lotus Notes*
   (b) durch SSL oder TLS gesicherte Datenkommunikation
   (c) einheitlicher Connector auf Host-System
   (d) Inventarisierung von mobilen Endgeräten inkl. Zuordnung zu bestimmten Benutzern (optional)
3. Kriterien für Sicherheitslösung
   (a) aktuelle Sicherheitsstandards
   (b) Kombination mit *Zwei-Faktor-Authentifizierung* (optional)
   (c) Einstellungen bis auf Hardware-Ebene (optional)
4. Kriterien für die Kosten
   (a) Höhe der Anschaffungskosten
   (b) Höhe der laufenden Kosten (Support, Lizenzen)
   (c) personelle Kosten

Für jede getestete Lösung wurde ein Bewertungsbogen erstellt, auf dem jedem Kriterium nach eine Note von 1 (= „sehr gut“ bzw. „trifft voll und ganz zu“) bis 5 (= „sehr schlecht“ bzw. „trifft überhaupt nicht zu“) zugewiesen ist. Dieser Bewertungsbogen soll neben der textlichen Testbeschreibung eine sachliche Betrachtung der einzelnen Lösung ermöglichen und dient einer besseren Vergleichbarkeit.

5.2 Die Test-Kandidaten

Nach Definition dieser Kriterien habe ich den Markt für mobile Lösungen auf genau diese Kriterien analysiert und habe nur ein paar Lösungen gefunden, die alle Hauptkriterien erfüllen. Dabei ist schon vorab anzumerken, dass es keine einheitliche Lösung für Datenkommunikation, also Synchronisation, Backup und Software-Verteilung, und Datensicherheit gibt. Ferner muss man genau diese beiden Gruppen unterscheiden. Auf der einen Seite gibt es sehr gute Lösungen, die ausschließlich für Datenkommunikation, Aufstellen von Benutzerregeln und Inventarisierung. Auf der anderen Seite ebenfalls sehr gute Lösungen für die Sicherheit der Daten. Eine komplette Lösung, die beide Möglichkeiten bietet, gibt es nicht.

Aus diesem Grund habe ich mir zwei wichtige Lösungen für den Bereich „Datenkommunikation, Regeln für Benutzer und Inventarisierung“ und zwei Lösungen für den Bereich

„Datensicherheit“ näher angeschaut. Für die Evaluierung dieser Lösungen hatte ich als Server einen *Compaq Evo D51S* mit einem 2 GHz *Pentium 4* Prozessor und 512 MB RAM. Als Betriebssystem habe ich *Windows XP Professional* verwendet. Als Clients hatte ich einige unterschiedliche Geräte. Als *Smartphone* hatte ich ein *Siemens SX1*[2] mit *Symbian OS* als Betriebssystem und ein *Motorola MPx200*[3] mit *Microsoft Windows SmartPhone 2002* als Betriebssystem. Als *PDA* kamen ein *Palm Zire* mit *Palm OS 4.1* und 2 MB Speicher, sowie ein *Dell Axim X5* mit *Microsoft PocketPC 2003* zum Einsatz. Als Laptop wurde mir von der *EDAG Engineering & Design AG* ein *Toshiba Satellite Pro 4600* mit einem 700 MHz *Pentium III* Prozessor und 256 MB RAM zur Verfügung gestellt. Die Verbindung zwischen den *PDAs* bzw. *Smartphones* und dem Synchronisations- bzw. Sicherheitsserver erfolgte immer über eine kabelgebundene Verbindung mit einem USB-Port des Laptops.

5.2.1 Datenkommunikation, Regeln für Benutzer und Inventarisierung

Betrachten wir zuerst den Bereich *Datenkommunikation, Regeln für Benutzer und Inventarisierung*. Beide gefundenen Lösungen sind sehr umfangreich in ihren Funktionen und bieten viele Möglichkeiten. Sie bieten beide den Zugriff auf die *Lotus Notes*-Groupware und sind damit ideale Kandidaten. Beide unterstützen die Verwaltung von *Palm OS*-, *Windows CE*- und *Symbian OS*-basierten *PDAs* und *Smartphones*. Auch die Verwaltung von Laptops ist mit beiden Lösungen möglich.

5.2.1.1 OneBridge

Die erste Lösung ist der *OneBridge*-Server von *Extended Systems*. Dieses Produkt konnte ich 60 Tage lang über eine Evaluierungslizenz genau testen. *OneBridge* läßt sich leicht installieren, wobei die Installationsanweisungen und der Installationsassistent der von mir getesteten Version 4.2.2004.721 nicht überein stimmen. Das ist nicht unerheblich, denn im Installationsassistenten sind die Komponenten, die für einen Zugriff auf *Lotus Notes* wichtig sind, an einer anderen Stelle als in der Installationsanweisung beschrieben. Ohne die Installation dieser Komponenten, der *Lotus Listener* und der *Lotus Adapter*, kann der *OneBridge*-Server nicht mit *Lotus Notes* kommunizieren, was eine Daten-Synchronisation der *PIM*-Daten zwischen mobilen Endgerät und der Groupware unmöglich macht. Die weitere Installation des *OneBridge*-Servers verlief aber reibungslos.

Nach der Installation starten sofort Assistenten, die alle grundlegenden Einstellungen abfragen, u.a. auch die Informationen für den Zugang zum *Lotus Notes*-Server. Dafür benutzt *OneBridge* den *Lotus Adapter* um auf einen lokal installierten *Lotus Notes*-Client über eine *Notes-ID* auf den *Lotus Notes*-Server zuzugreifen. Die *Notes-ID* dient dabei zur Authentifizierung des Benutzers gegenüber dem *Lotus Notes*-Server. *OneBridge* bietet darüber hinaus

[2] wurde mir von *Siemens Mobile* für meine Diplomarbeit zur Verfügung gestellt
[3] wurde mir von *Motorola* für meine Diplomarbeit zur Verfügung gestellt

Abb. 5.1: *Assistent für die Verbindung zur Groupware*

zwei Möglichkeiten der Authentifizierung. Zum einen kann für jeden berechtigten Benutzer eine eigene *Notes-ID* hinterlegt werden. Dies ist sehr unpraktikabel, da es hier zu Unterschieden zwischen einer z.B. lokal auf einem *Laptop* installierten *Notes-ID* und der auf dem *OneBridge*-Server geben kann, insbesondere wenn einmal das Passwort geändert wurde. Diese Daten immer aktuell zu halten ist nur mit entsprechender Anstrengung zu erreichen und Probleme sind quasi vorprogrammiert. Besser ist das Anlegen einer eigenen Benutzergruppe am *Lotus Notes*-Server, in der alle Benutzer hinterlegt sind, die Berechtigung haben auf den *OneBridge*-Server zuzugreifen. Hier wird auf dem *OneBridge*-Server eine eigene *Notes-ID* hinterlegt, die ausschließlich lesenden Zugriff auf die *Lotus Notes*-Benutzerdatenbank hat und überprüft, ob der anfragende Benutzer in dieser Gruppe ist und ob seine Anmeldedaten stimmen. Wurde dieser Verbindungsassistent abgeschlossen, so sind auch schon alle wichtigen Einstellungen getan. Die Einstellmöglichkeiten der anderen Assistenten sind recht selbsterklärend und werden deshalb hier nicht erwähnt. Nach einem Neustart von *Windows* ist der *OneBridge*-Server einsatzbereit.

Startet man den *OneBridge Sync Admin*, so wird man zunächst gefragt, welche Konfiguration man bearbeiten will. Standardmäßig steht hier die *Starter.xcf* zur Verfügung. Das ist genau die Konfiguration, die man während der Installationsphase erstellt hat. Hat man die Konfigurationsdatei ausgewählt, werden alle Einstellungen geladen und man erhält den Zugriff auf die Administrationsoberfläche. Der Aufbau der Oberfläche erinnert an die *Microsoft Management Console*, die z.B. Grundlage für die *Computerverwaltung* von *Windows XP* ist. Auf der linken Seite hat man eine Baumstruktur der einzelnen Module und auf der rechten Seite erhält man zusätzliche Informationen. Um eine Einstellung vorzunehmen, ist ein Doppelklick auf den jeweiligen Eintrag notwendig, worauf sich ein Dialog-Fenster öffnet, in dem man die Einstellungen vornehmen kann. Diese Baumstruktur macht einen aufgeräumten Eindruck, jedoch ist negativ anzumerken, dass nach dem Laden der Konfigurationsdatei die Baumstruktur komplett aufgeklappt ist und man schnell von der Menge der angezeigten Daten erschlagen wird. Besser wäre aus meiner Sicht ein komplett zugeklappte Baumansicht und der Benutzer kann selbst die Unterpunkte öffnen, die momentan für ihn wichtig sind. Der *OneBridge*-Server bietet dabei eine Menge an Einstellungen, die auf unterschiedliche Systeme aufgeteilt sind. Auf einen *OneBridge*-Server können mobile Endgeräte mit *Windows CE*, *Palm OS*, *Symbian OS*, aber auch einer Desktop-Version von *Microsoft*

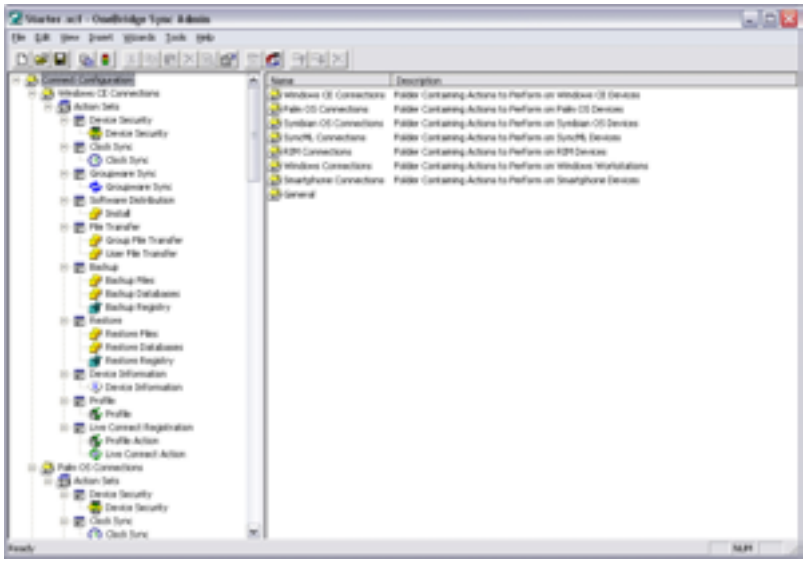


Abb. 5.2: *OneBridge Sync Admin*

Windows zugreifen. Auch die Unterstützung von Endgeräten mit *SyncML* ist gegeben. Außerdem unterscheidet *OneBridge* nochmal explizit den Zugriff von *Smartphones*. Hier sind speziell Endgeräte mit *Windows Smartphone 2002* gemeint. Für jedes dieser Systeme können nun eine Menge an Einstellungen vorgenommen werden. Was wird z.B. mit der Groupware synchronisiert, welche Software soll installiert werden, wann soll ein Backup durchgeführt werden usw. Eigentlich sind alle Einstellungen vorhanden, die man sich wünschen kann. Allgemein deckt der *OneBridge*-Server alle Möglichkeiten der Datensynchronisation mit einer Groupware, dem Backup und Restore eines mobilen Endgeräts und der Software- und Dateiverteilung ab. *OneBridge* bietet sogar eine grundlegende Einstellung für die Sicherheit des mobilen Endgerätes. Diese Einstellung überprüft jedoch nur, ob auf dem mobilen Endgerät der Passwortschutz aktiviert ist, oder nicht. Ist er nicht aktiviert, so verweigert *OneBridge* den Zugriff auf den Server und die Synchronisation der Daten.

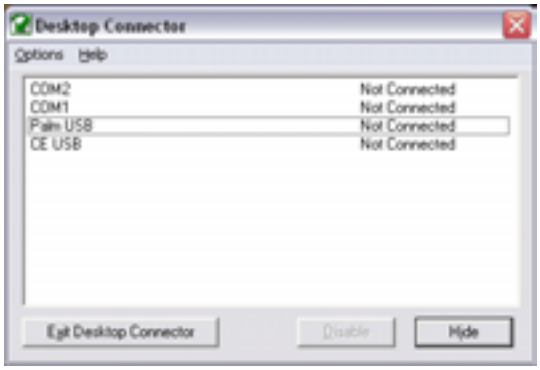


Abb. 5.3: *OneBridge Desktop Connector*

Damit ein Client überhaupt eine Verbindung zum *OneBridge*-Server aufbauen kann, benötigt er eine kleine Software die dies ermöglicht. Im Prinzip entspricht diese Software *ActiveSync* oder *HotSync* und ist genau so zu bedienen. Jedoch ist es die gleiche Software für alle Systeme. Wenn man, wie in meiner Testumgebung, die Verbindung über einen Host-Rechner über ein USB-Kabel aufbauen will, so muss auf dem Host-Rechner der *OneBridge Desktop Connector* installiert sein. Der *OneBridge Desktop Connector* bietet nur die Möglichkeit ein über USB angeschlossenes Endgerät den Zugang zum Netzwerk und somit zum *OneBridge*-Server zu ermöglichen. Auch das ist ähnlich zu *ActiveSync* oder *HotSync*. Der *OneBridge Desktop Connector* ist jedoch für jedes mobile Endgerät einsetzbar und macht keinen Unterschied, ob es sich z.B. um ein *Windows CE*- oder *Palm OS*-Gerät handelt. *ActiveSync* und *HotSync* überprüfen vorher, welches mobile Endgerät nun angeschlossen ist. Außerdem bieten *ActiveSync* und *HotSync* noch zusätzlich die Möglichkeit, dass der Benutzer selbstständig Software darüber installiert oder sein mobiles Endgeräte mit einem lokal installierten *Microsoft Outlook* z.B. seine privaten Daten synchronisiert. Diese Möglichkeit bietet der *OneBridge Desktop Connector* nicht. Die Installation sowohl des *OneBridge Desktop Connectors*, als auch der Client-Software auf den mobilen Endgeräten verlief ohne Probleme.

Die Bedienung des *OneBridge*-Clients auf dem mobilen Endgerät ist weitest gehend selbsterklärend und einem Benutzer in kurzen Worten erklärt. Ist der *OneBridge Desktop Connector* einmal installiert, muss an diesem nichts mehr eingestellt werden, sondern er erkennt automatisch, wenn ein mobiles Endgerät über eine serielle Schnittstelle angebunden wurde. Alle angeschlossenen Testgeräte wurden sofort zum *OneBridge*-Server weiterverbunden. Dafür zuständig sind die auf dem mobilen Endgerät hinterlegten Profile, in denen alle wichtigen Verbindungseinstellungen für den Zugriff auf den *OneBridge*-Server hinterlegt sind. Bei allen Testgeräten hat dieser Verbinungsaufbau ohne Probleme funktioniert, egal ob *Symbian-Smartphone* oder *Windows-PDA*. Einzig die eigentliche Synchronisation mit der Groupware hat nicht funktioniert, was aber auf die Einschränkungen der Evaluierungslizenz zurückzuführen ist. Alles andere funktionierte wie erwartet. Der Server verweigerte den Zugriff, wenn die Passwort-Funktion nicht aktiviert wurde und schickte die Daten an das jeweilige Endgerät, für das sie bestimmt waren. Nur der *Palm Zire* brach die Übertragung ab, was laut Log-Datei an der geringen Speichererweiterung von nur 2 MB lag. Die Einbindung des *OneBridge*-Servers in ein bereits vorhandenes Netzwerk ist ohne Probleme durchzuführen. Ein externer Zugriff, beispielsweise über eine VPN-Verbindung zum *EDAG*-Netzwerk, wurde in diesem Test nicht durchgeführt. Standardmäßig ist die externe Kommunikation zwischen dem *OneBridge*-Server und dem mobilen Endgerät über *SSL* bzw. *TLS* geschützt.

Der *OneBridge*-Server erfüllt somit alle geforderten Kriterien. Er ist leicht zu installieren und zu bedienen, bietet eine Synchronisation mit *Lotus Notes* und unterstützt alle aktuellen mobilen Endgeräte. Die Anschaffungskosten der *OneBridge*-Lösung schmälern leider etwas den guten Eindruck. Es wird nach *Gerätelizenzen* abgerechnet, d.h. für jedes Gerät, das auf den *OneBridge*-Server zugreifen will, muss eine Lizenz erworben werden. Die Kosten liegen hier bei 185,- € pro Gerätelizenz. An weiteren Support- und Lizenzkosten müssen pro Jahr 20% des tatsächlichen Kaufpreises aufgewendet werden. Darin enthalten sind alle Software-Updates und die externe Unterstützung per eMail und/oder Telefon. Für die Installation sind ca. 2 Manntage notwendig, die unbedingt mit externer Unterstützung ablaufen sollte. Darin enthalten ist auch eine kurze Einweisung eines Administrators, um alle Einstellungen am *OneBridge*-Server vornehmen zu können.

Alle Bewertungen sind noch einmal in der Tabelle 5.1 zusammengefasst.

Allgemeine Kriterien

	1	2	3	4	5
einheitliche Administrierung		×			
Administrierung mobiler Systeme			×		
Einstellungen für Benutzer vordefinierbar			×		
einfache Handhabung für Administratoren und Benutzer	×				
Kopplung mit Benutzerverwaltung		×			
Integrierbarkeit in Netzwerk	×				
Kombinierbar (optional)		×			

Spezielle Kriterien für Datenkommunikation

	1	2	3	4	5
Synchronisation mit Lotus Notes		×			
Verbindungsschutz durch SSL/TLS		×			
einheitlicher Connector	×				
Inventarisierung (optional)		×			

Kriterien für Kosten

	1	2	3	4	5
Anschaffungskosten			×		
Support-Kosten			×		
personelle Kosten		×			

Tabelle 5.1: *Bewertungsbogen OneBridge*

5.2.1.2 Pylon Anywhere

Eine weitere Lösung für den Bereich *Datenkommunikation, Regeln für Benutzer und Inventarisierung* ist der *Pylon Anywhere*-Server von *Sybase*. Leider war nicht mehr genügend Zeit, um eine vollständige Evaluierung dieser Lösung durchzuführen. Alle Angaben beziehen sich daher auf eine Live-Demonstration durch die Firma *ECOPLAN*, Fulda, sowie den Produktbeschreibungen der Lösung.

Der *Pylon Anywhere*-Server bietet eigentlich alles, was auch schon bei der *OneBridge*-Lösung realisiert wurde. Er kann ohne Probleme mit der *Lotus Notes*-Groupware verbunden werden und unterstützt dabei die mobilen Betriebsysteme *Palm OS*, *Windows CE* und *Symbian OS 7*, sowie die Desktop-Versionen von *Microsoft Windows*, aber auch *SyncML*-fähige Endgeräte. Zur Synchronisation zwischen dem *Pylon Anywhere*-Server und den mobilen Endgeräten ist ebenfalls eine eigene Software notwendig, die auf dem zugreifenden Endgerät installiert sein

muss. Diese Synchronisations-Software ersetzt zumindest auf dem mobilen Endgerät *HotSync* oder *ActiveSync*, auf einem Laptop sind beide jedoch notwendig, um einen *Palm* bzw. *PocketPC* über eine serielle Schnittstelle anzuschließen. Es existiert hier kein einheitlicher *Desktop Connector*, wie es bei *OneBridge* der Fall ist. Wenn man sowohl ein *Palm OS*-, als auch ein *Windows CE*-Endgerät über ein Host-System anbinden möchte, so muss auf dem Host-System sowohl *ActiveSync* für den *PocketPC*, als auch *HotSync* für den *Palm* installiert sein. Somit hat auch der Benutzer wieder die Möglichkeit evt. unerlaubter Weise sein mobiles Endgerät mit privaten Daten zu füllen. Möchte man diese Lösung einsetzen, so ist es bestimmt sinnvoll eine zusätzliche Software zu finden, die dem *OneBridge Desktop Connector* gleichzusetzen ist.

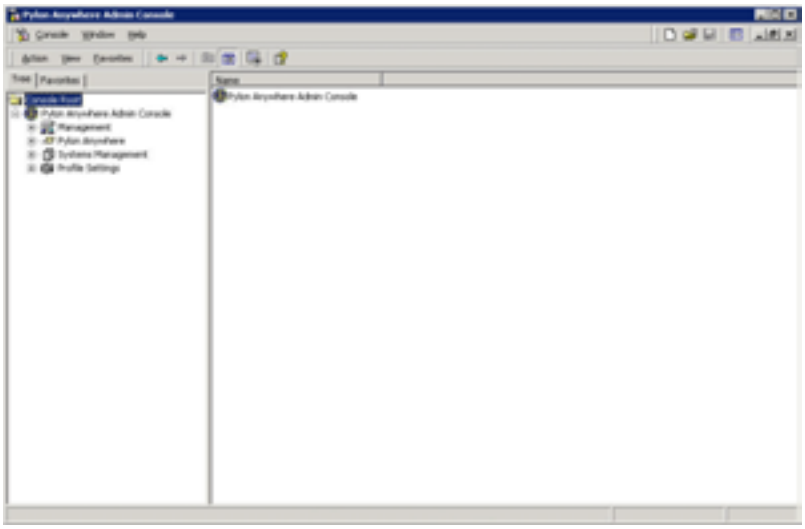


Abb. 5.4: *Pylon Anywhere Admin Console*

Die *Pylon Anywhere Admin Console* zeigt sich sehr aufgeräumt und kann direkt in die *Microsoft Management Console* integriert werden. Es ist somit möglich die *Pylon Anywhere Admin Console* in eine bestehende Konfiguration einer *Microsoft Management Console*, die beispielsweise für die Administrierung eines *Microsoft Active Directory* zuständig ist, einzubinden. Die *Pylon Anywhere Admin Console* macht einen übersichtlichen Eindruck, da wieder die übliche Baumstruktur im linken Teil verwendet wird. Nach einem Start der Administratoroberfläche bleiben alle Wurzelelemente geschlossen und der Administrator wählt selbst die Unterbereiche, die momentan für ihn interessant sind. Auf der rechten Seite erscheinen, nach der Auswahl eines Elementes auf der linken Seite, weitere Informationen zu dem jeweiligen Unterpunkt. Die Benutzerverwaltung kann über *LDAP* an ein bestehendes Benutzerverzeichnis angebunden werden, was die Integration bestehender Benutzer sehr vereinfacht. *Pylon Anywhere* speichert nur die Informationen, die über das *LDAP-Verzeichnis* nicht abgedeckt werden. Die *Pylon Anywhere Admin Console* bietet sehr viele Einstellmöglichkeiten, die alle auf einen Benutzer bzw. eine Benutzergruppe bezogen sind. Wenn ein Benutzer mal ein anderes Endgerät benutzt und z.B. zwischen einem *PocketPC* und einem *Palm* wechselt, dann gelten für beide Geräte die gleichen Einstellungen, ohne dass der Administrator weitere Einstellungen vornehmen muss. Auf dem Endgerät muss lediglich die *Pylon*-Client-Software mit dem Profil des Benutzers installiert sein und alles funktioniert für den Benutzer wie gewohnt. Der Administrator kann in diesem Profil alle Informationen

für die Verbindungsmöglichkeiten zum *Pylon Anywhere*-Server hinterlegen, aber auch definieren, inwiefern der Benutzer die Einstellungen selbst verändern darf. Funktionalitäten für eine Software- oder Datei-Verteilung ist allerdings nicht standardmäßig im *Pylon Anywhere*-Server enthalten und muss über das Addon *File Management* zusätzlich gekauft werden. Auch eine direkte Verwaltung der mobilen Endgeräte inkl. Inventarisierungsmöglichkeiten werden über das zusätzliche Addon *System Management* abgebildet. Mit beiden Addons als Beigabe steigert sich aber die Funktionalität des *Pylon Anywhere*-Servers erheblich.

Für die Sicherheit der Verbindung zwischen dem Server und einem mobilen Endgerät steht nicht mehr nur *SSL* bzw. *TLS* zur Verfügung, sondern auch *3DES* (mit 112 Bit Schlüssellänge) und *AES* (mit 128 Bit Schlüssellänge). Auch die Möglichkeiten der Authentifizierung steigert sich z.B. durch die Integration einer *Zwei-Faktor-Authentifizierung*. Sollte ein Endgerät einmal gestohlen werden oder verloren gehen, so hat man zusätzlich die Möglichkeit der Server anzuweisen, dass bei der nächsten Synchronisation alle Daten auf dem mobilen Endgerät gelöscht werden sollen. Somit ist zwar das Gerät verloren, aber nicht unbedingt die Daten in falschen Händen. Darüber hinaus bietet diese erweiterte Version des *Pylon Anywhere*-Servers detaillierte Informationen über die einzelnen mobilen Endgeräte. So werden beispielsweise alle verfügbaren Hardware-Informationen, also *WLAN*, *Bluetooth*, *Hauptspeicher*, *Prozessor* usw., wie auch die installierte Software inkl. deren Versionen ausgelesen und zentral gespeichert. Diese Informationen dienen aber nicht nur der Inventarisierung, sondern können auch verwendet werden, wenn die Skripting-Funktion von *Pylon Anywhere* genutzt werden soll. Über eigene Skripte können dadurch Regeln aufgestellt werden, ob ein bestimmter Typ von Endgerät, z.B. nur *Palm OS*-Geräte, bestimmte Software zugeteilt bekommt. Diese Skripte werden dann während der Synchronisationsphase abgearbeitet und die darin enthaltenen Anweisungen ausgeführt. Die verwendete Skriptsprache sieht sehr einfach aus und dürfte innerhalb kürzester Zeit von einem Administrator erlernbar sein. *Pylon Anywhere* bietet durch die Erweiterungen auch eine Vielzahl an Möglichkeiten der Auswertung. So können Berichte erstellt werden, wann welcher Benutzer wie oft synchronisiert hat, oder welche Fehler sich bei einer Synchronisation ergeben haben. Diese Informationen sind vor allem wichtig, wenn sich Probleme ergeben und ein Benutzer mit dem IT-Support im Unternehmen Kontakt aufnimmt. Alle Daten sind dabei exportierbar oder über HTML-Seiten darstellbar. Hier mit dazu gehört auch das Backup-Management des *Pylon Anywhere*-Servers. Zuerst definiert ein Administrator wie oft ein Backup des Gerätes angefertigt werden soll. Das Backup wird dabei verschlüsselt auf dem *Pylon Anywhere*-Server abgelegt. Sollte es einmal zum Verlust der Daten kommen, z.B. weil die Batterie des Endgerätes leer ist, so kann der IT-Support eine Webseite bereitstellen, über die der Benutzer das letzte Backup seines Endgerätes wieder zurückspielen kann. Zum Schluß bietet der *Pylon Anywhere*-Server noch die Möglichkeit einer Webmail-Oberfläche. Dazu wird über eine Internet-Adresse eine einheitliche Oberfläche hzur Verfügung gestellt, über die ebenfalls alle *PIM*-Daten eingesehen und bearbeitet werden. Die Verbindung zu dieser Webmail-Oberfläche sollte aber durch *SSL* bzw. *TLS* gesichert werden.

Interessant ist auch auf jedenfall das Lizenzmodell der *Pylon Anywhere*-Lösung, da hier nach Benutzern abgerechnet wird. Somit wird jeweils nur für einen Benutzer bezahlt, wobei dieser bis zu fünf mobile Endgeräte gleichzeitig mit dem *Pylon Anywhere*-Server nutzen darf. Dies ist auf jeden Fall ausreichend, da in der Regel kein Benutzer wirklich fünf Endgeräte gleichzeitig benutzt, sondern evt. maximal drei (*PDA*, *Mobiltelefon* und *Laptop*). Die Anschaffungskosten betragen für den *Pylon Anywhere*-Server inkl. den Addons *File Management* und *System Management* 169,- € pro User. Vergleicht man die Funktionalitäten

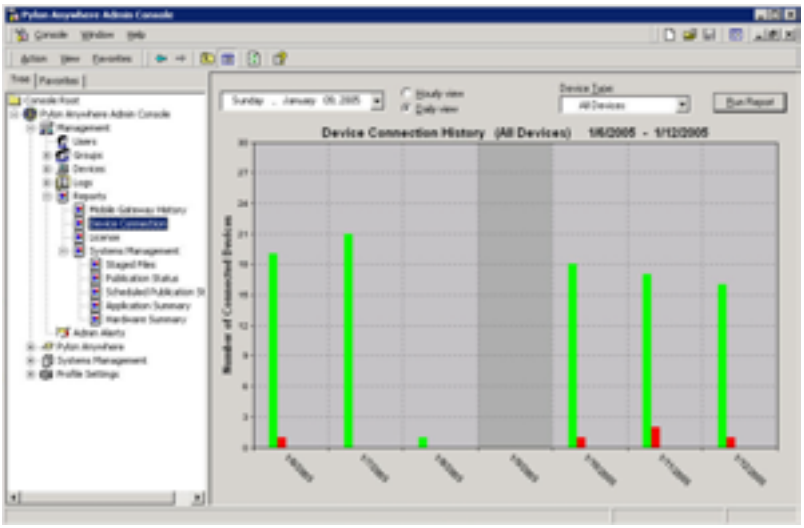


Abb. 5.5: *Pylon Anywhere Reporting Ansicht*

des *Pylon Anywhere*- mit dem *OneBridge*-Server, so ist *Pylon Anywhere* deutlich günstiger. An laufenden Support- und Lizenz-Kosten sind 34,- € pro Benutzer und Jahr fällig, wobei dieser Betrag bereits bei der Anschaffung fällig ist und wie schon beim *OneBridge*-Server ca. 20% des Anschaffungspreises entspricht. Darin enthalten sind alle Software-Updates und Support über Telefon und/oder eMail. Da die *Pylon Anywhere*-Lösung zusätzlich von einem ortsansässigen Unternehmen angeboten wird, ist bei größeren Problemen mit einer recht schnellen Reaktion evt. auch vor Ort bei der *EDAG Engineering & Design AG* zu rechnen sein. Für die Installation ist wieder mit ca. 2 Manntagen zu rechnen, die mit externer Unterstützung ablaufen sollte. Die Integration in das bestehende Netzwerk der *EDAG Engineering & Design AG* stellt hier ebenfalls kein Problem dar. Bei der Installation ist auch eine ausführliche Einweisung für einen Administrator enthalten. In wieweit ein Handbuch bzw. eine Online-Hilfe Unterstützung bieten, kann an dieser Stelle nicht beurteilt werden. Auch hier gilt: Sind einmal alle Einstellungen vorgenommen, so muss der Administrator bei einem neuen Gerät nur noch die benötigte Software mit einem gültigen Benutzerprofil installieren und alles läuft.

Alle Bewertungen sind noch einmal in der Tabelle 5.2 zusammengefasst.

5.2.2 Datensicherheit

Aktuelle mobile Endgeräte bieten nur wenig Möglichkeiten, die auf dem Endgerät gespeicherten Daten zu schützen. In den meisten Fällen kann das Endgerät über ein einfaches Passwort vor unerlaubten Zugriffen geschützt werden, dies ist aber standardmäßig kaum aktiviert. Eine Verschlüsselung der Daten sucht man meist vergebens. Dazu ist in der Regeln zusätzliche Software nötig, die diese Aufgaben übernimmt. Von dieser Art von Software gibt es derzeit jedoch eine Menge, mit dem Problem, dass sie immer nur ein mobiles System unterstützen,

Allgemeine Kriterien

	1	2	3	4	5
einheitliche Administrierung		×			
Administrierung mobiler Systeme		×			
Einstellungen für Benutzer vordefinierbar	×				
einfache Handhabung für Administratoren und Benutzer	×				
Kopplung mit Benutzerverwaltung		×			
Integrierbarkeit in Netzwerk	×				
Kombinierbar (optional)		×			

Spezielle Kriterien für Datenkommunikation

	1	2	3	4	5
Synchronisation mit Lotus Notes	×				
Verbindungsschutz durch SSL/TLS	×				
einheitlicher Connector				×	
Inventarisierung (optional)		×			

Kriterien für Kosten

	1	2	3	4	5
Anschaffungskosten			×		
Support-Kosten			×		
personelle Kosten		×			

Tabelle 5.2: *Bewertungsbogen Pylon Anywhere*

also nur *Windows CE* oder nur *Palm OS*. Nur wenige unterstützen mehrere Systeme und zwei davon konnte ich testen.

5.2.2.1 Pointsec

Die erste Lösung ist die Sicherheitslösung von *Pointsec*. *Pointsec* ist ein schwedisches Unternehmen und derzeit Weltmarktführer von Sicherheitslösungen für mobile Endgeräte. *Nokia* will zukünftig die *Pointsec*-Lösungen für seine *Symbian OS*-Geräte einsetzen, wobei zunächst der neue *Nokia 9500 Communicator* unterstützt wird[4].

Die *Pointsec*-Lösung unterteilt sich in verschiedene Einzelprodukte, die jeweils einzeln betrachtet ein mobiles System unterstützen. Es gibt *Pointsec for Palm OS*, *Pointsec for PocketPC*, *Pointsec for Symbian* und *Pointsec for Smartphone*. Zur Evaluierung standen mir *Pointsec for Symbian* in Version 2.0 und *Pointsec for PocketPC* in Version 2.3 60 Tage lang zur Verfügung. Darüber hinaus bietet *Pointsec* auch eine Verschlüsselung von PCs, auch *Pointsec for PC* genannt. Dieses Produkt wurde mir durch eine Live-Demonstration von *Pointsec* auf der *SYSTEMS 2004* in München vorgestellt.

Die Installation der *Pointsec*-Lösungen gestaltet sich recht einfach. Es ist nur das Installationsprogramm auszuführen und allen Anweisungen zu folgen. Dabei muss zunächst die *Pointsec Administration Console* und dann alle weiteren Produkte einzeln installiert werden. Um die Einstellungen für die Produkte vorzunehmen, ist später nur noch die Administrationsoberfläche notwendig. Der Zugriff auf diese Oberfläche ist durch ein spezielles *Challenge-Response-Verfahren* geschützt. Dabei werden über den *Pointsec Account Manager* Benutzer angelegt und diesen ein sog. *Token*[5] zugewiesen. Dieser *Token* dient dazu, den entsprechenden *Response* zu einer dazugehörigen *Challenge* zu generieren. Diese Aufgabe übernimmt die Software *Pointsec X9.9-Token*. Darin wählt man seinen *Token* aus und identifiziert sich über eine *PIN*. Dadurch hat man Zugriff auf die Eingabe des *Challenge*. *Pointsec X9.9-Token* generiert darauf einen *Response*, der bei der Authentifizierung, z.B. bei der Anmeldung an der *Pointsec Administration Console*, eingesetzt werden kann. Dieses Verfahren ist zwar recht umständlich, bietet aber die maximale Sicherheit für den Zugang auf die *Administrator Console*.

Abb. 5.6: *Pointsec X9.9-Token*

Der Aufbau der Administrationsoberfläche gestaltet sich wieder nach dem gewohnten Bild. Auf der linken Seite eine Baumstruktur mit einigen Wurzelelementen und auf der rechten Seite die nötigen Informationen und Möglichkeiten für Einstellungen. Als Hauptelemente werden hierbei die installierten Produkte angezeigt, in meinem Fall also *Pointsec for PocketPC* und *Pointsec for Symbian*, gefolgt von zwei bzw. drei Unterpunkten. Bei beiden gibt es die Auswahl zwischen *Create Installation Set* und *Remote Help*, bei *Pointsec for*

[4] *siehe* http://www.pr-com.de/PRI.nsf/0/C77ECBD0B60D8DB1C1256EFD0045E9F4

[5] Dieser *Token* ist nicht mit dem in Kapitel 3.1.2 beschriebenen *Token* zu verwechseln.

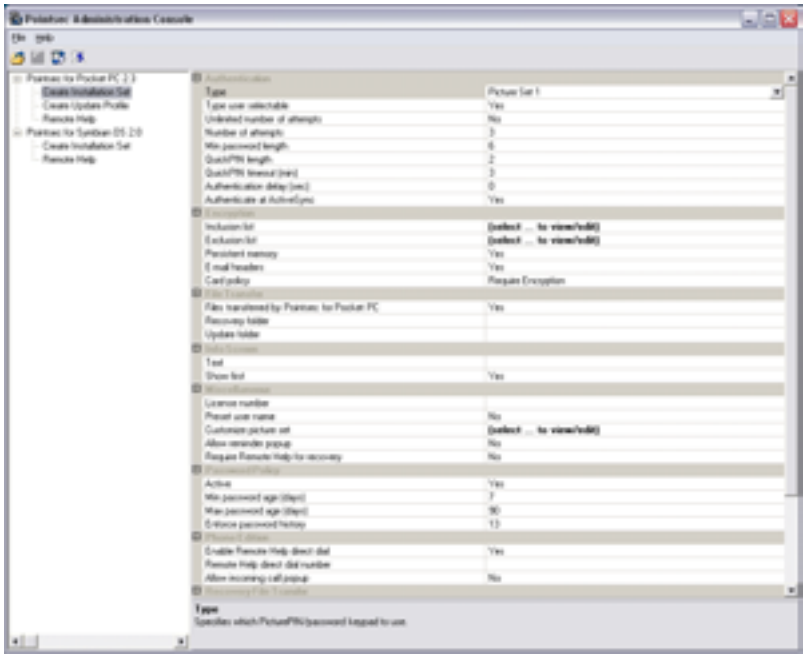


Abb. 5.7: *Pointsec Admin Console*

PocketPC zusätzlich *Create Update Profile*. Letzteres dient eigentlich nur dazu ein bestehendes Profil zu aktualisieren. Für den ersten Schritt ist *Create Installation Set* wichtig. Dort werden alle grundlegenden Einstellungen gesetzt, die für das jeweilige Gerät gelten sollen. Beispielsweise die Art der Authentifizierung, welche Dateien und Verzeichnisse verschlüsselt werden sollen und welche nicht, eine kleine *Password Policy*, was bei einem Dateitransfer bzw. einem Recovery des Gerätes passiert und nach welcher Zeit der Bildschirm gesperrt wird. Dabei kann die Menge der Einstellungen zwischen den Produkten sehr unterschiedlich sein. Bei *Pointsec for Symbian* begrenzt sich die Anzahl der Optionen auf einige wenige, wie Authentifizierung und der Verschlüsselungspolicy. Alle Einstellungen sind sehr übersichtlich gestaltet und durchaus selbsterklärend. Für die Authentifizierung bietet *Pointsec* neben einem alphanumerischen Passwort und einer *PIN* noch eine zusätzliche Möglichkeit, *PicturePIN* genannt. Dabei werden dem Benutzer zur Anmeldung neun Bilder angezeigt, auf die dieser nach einer festgeklickten Reihenfolge mit dem Stift tippen muss, um sich an dem mobilen Endgerät anzumelden. Dazu können zwei vordefinierte Bildersets aber auch eigene Bilder eingesetzt werden. Hat man z.B. eingestellt, dass das Passwort vier Zeichen bzw. Bilder lang sein muss, so denkt man sich evt. einen kleinen Satz aus, in dem vier der möglichen Bilder vorkommen und bildet daraus sein Passwort. Sind beispielsweise die Bilder „Logo der *EDAG*“, „Mann in Anzug“, ein „Brief“ und ein „Laptop“ vorhanden[6], so

Abb. 5.8: *Pointsec PicturePIN*

[6] *siehe* Abbildung 5.8

könnte der dazu gehörige Satz „Bei der *EDAG* schreibe *ich* viele *eMails* auf meinem *Laptop*“ (Passwort = „Logo“ – „Mann“ – „Brief“ – „Laptop“) lauten. Dabei wechseln die Bilder bei jeder neuen Anmeldung, um zu verhindern, dass ein Angreifer alleine durch Abschauen der Positionen des Stiftes bei der Eingabe das Passwort errät. Bei den Einstellungen zur Verschüsselung kann genau definiert werden, welche Verzeichnisse bzw. welche Dateien geschützt werden sollen und welche nicht. Das geht sogar soweit, dass die Daten im Hauptspeicher verschlüsselt werden können. Des weiteren kann man definieren, wie mit mobilen Speicherkarten umzugehen ist. Hier kann man zwischen *Require Encryption* und *No Restrictions* auswählen. Als Verschlüsselungsalgorithmus wird stets *AES* mit einer Schlüssellänge von 128 Bit eingesetzt. Sind alle Einstellungen vorgenommen erstellt man das *Installation Set* und überträgt es auf das mobile Endgerät. Dazu kann beispielsweise eine der oben beschrieben Lösungen für Software- und Daten-Verteilung benutzt werden. In beide Lösungen läßt sich *Pointsec* ohne großen Aufwand integrieren. Darüber können Updates der Profile verteilt, aber auch *Recovery Files* des Endgerätes für eine spätere Wiederherstellung gespeichert werden.

Nach der Installation von *Pointsec* auf dem mobilen Endgerät merkt der Benutzer nicht viel von den Veränderungen. Lediglich die Anmeldeprozedur am Endgerät ändert sich, aber die Bedienung wird dadurch nicht erschwert. Die Ver- und Entschlüsselung geschieht in einer akzeptablen Zeit. Sollten natürlich Dateien von mehreren hundert MB ver- und entschlüsselt werden, dann ist mit einer Wartezeit zu rechnen. Die Authentifizierung über *PicturePIN* ist sehr einfach in der Handhabung und bietet einen schnellen Zugriff auf das Endgerät. Man hat zumindest nicht mit Problemen bei der Schrifterkennung zu kämpfen, was ja bei einem alphanumerischen Passwort eher passieren kann. Wenn man nur schnell auf dem *PDA* nachschauen will, welchen Termin man als nächstes hat, so ist der Einsatz von *PicturePIN* eine prima Sache. Im Profil kann zusätzlich die Anzahl der möglichen Fehlversuche bei der Authentifizierung hinterlegt werden. Hat man diese Anzahl überschritten, so wird das Endgerät automatisch gesperrt. Um das Endgerät wieder entsperren zu lassen, bietet die *Pointsec Administration Console* für jedes Produkt die Möglichkeit *Remote Help*. Der Benutzer meldet sich dazu beispielsweise beim IT-Support des Unternehmens und generiert auf seinem Gerät eine *Challenge*. Der Support-Mitarbeiter lädt in der Administrationsoberfläche von *Pointsec* das *Recovery File* des gesperrten Endgerätes und trägt die mitgeteilte *Challenge* ein. Weiterhin muss noch gewählt werden, ob das Gerät entsperrt oder die *Pointsec*-Software entfernt werden soll. Gemäß diesen Informationen wird ein *Response* berechnet, den der Benutzer nur noch in sein Endgerät eintragen muss. Somit ist der Zugriff auf das Gerät wieder hergestellt.

Pointsec for PC bietet weitaus mehr Möglichkeiten um einen Laptop oder auch einen Desktop-PC zu schützen. Dabei wird aber nur *Microsoft Windows* als Betriebssystem unterstützt. *Pointsec for PC* verfügt über eine eigene Benutzerverwaltung, über die definiert werden kann, wer mit der Software ausgestattet wird und wer nicht. Als Grundlage für diese Verwaltung kann ein *Microsoft Active Directory* verwendet werden. So erspart man sich das aufwendige Neuanlegen aller berechtigten Benutzer. Für jeden Benutzer bzw. jede Benutzergruppe werden nun Profile angelegt, in denen alle benötigten Einstellungen hinterlegt sind. Hier sind natürlich viel mehr Einstellungen möglich, als für einen *PDA* oder ein *Smartphone*. Bei der Authentifizierung wird hier beispielsweise eine *Zwei-Faktor-Authentifizierung*, z.B. mit *SafeWord* der Firma *SecureComputing*, unterstützt, aber auch der Einsatz von *Smartcards* oder *Tokens*[7]. Beim Verschlüsselungsalgorithmus hat man die Möglichkeit sich zwischen *AES* mit einer Schüssellänge von 256 Bit und *3DES* auszuwählen. Dabei kann eingestellt werden, ob eine komplette Verschlüsselung der Festplatte oder nur eine Verschlüsselung

[7] *siehe* Kapitel 3.1.2 auf Seite 20

von einzelnen Dateien und Verzeichnissen durchgeführt werden soll. Wie sich die Ver- und Entschlüsselung der Daten im laufenden Betrieb auf einem Laptop auf die Performance auswirkt, ist aber noch genau zu testen. Ob die PC-Lösung sinnvoll ist, muss über eine weitere Evaluierung nochmal im Detail geprüft werden. *Pointsec for PC* ist gemäß den *Common Criteria*[8] nach dem *Evaluation Assurance Level 4* zertifiziert[9].

Als Lizenzmodell bietet *Pointsec* eine Benutzerlizenz mit einem Anschaffungspreis von 150,- € pro Lizenz an. Bei dieser Lizenz kann ein Benutzer bis zu drei Produkte von *Pointsec* einsetzen. Soll bei einem Benutzer beispielsweise ein Laptop und ein *PDA* geschützt werden, so bietet sich diese Lizenz an. Alternativ kann jedes Produkt natürlich auch einzeln erworben werden, wobei die Lösungen für einen *PDA* 56,- € pro Lizenz und die PC-Lösung 110,- € pro Lizenz kosten. Die jährlichen Kosten für Support und Software-Upgrade und Maintenance belaufen sich dann zusätzlich auf 20% der kompletten Lizenzkosten. Für die Installation und Einrichtung der Lösungen gibt *Pointsec* als Dauer einen Manntag an. Dabei wird alles von *Pointsec* erbracht und eine Einweisung von einem Administrator für das System vorgenommen. Wenn eine ausführliche Schulung gewünscht wird, so ist mit einem weiteren Manntag zu rechnen. An den *Pointsec*-Lösungen muss dann ebenfalls nichts mehr gemacht werden. Für alle Arten von Endgeräten werden einheitliche Profile erzeugt und beispielsweise über den *Pylon Anywhere*-Server zentral verteilt. Lediglich bei der PC-Lösung sind evt. für jeden neuen Benutzer oder jedes neue Gerät eigene Einstellungen nötig. Das muss durch eine genaue Evaluierung noch festgestellt werden.

Alle Bewertungen sind noch einmal in der Tabelle 5.3 zusammengefasst.

[8] *siehe* Kapitel 6.1.1 auf Seite 74
[9] *siehe* http://www.pointsec.de/products/products_cert.asp

Allgemeine Kriterien

	1	2	3	4	5
einheitliche Administrierung			×		
Administrierung mobiler Systeme			×		
Einstellungen für Benutzer vordefinierbar			×		
einfache Handhabung für Administratoren und Benutzer			×		
Kopplung mit Benutzerverwaltung		×			
Integrierbarkeit in Netzwerk		×			
Kombinierbar (optional)			×		

Spezielle Kriterien für Sicherheitslösung

	1	2	3	4	5
Aktuelle Sicherheitsstandards	×				
Zwei-Faktor-Authentifizierung (optional)		×			
Detaillierte Sicherheitseinstellungen (optional)			×		

Kriterien für Kosten

	1	2	3	4	5
Anschaffungskosten			×		
Support-Kosten			×		
personelle Kosten		×			

Tabelle 5.3: *Bewertungsbogen Pointsec*

5.2.2.2 Trusted Mobility Server

Zweite interessante Lösung ist der *Trusted Mobility Server* von *Trust Digital*. Der *Trusted Mobility Server* ist Teil der *TRUST Enterprise Secure*-Lösung. Darin enthalten ist neben dem eigentlichen Sicherheitsmanagement für mobile Enderäte auch ein Informationsmanagement über die einzelnen Geräte. Dadurch kann man aus dem mobilen Endgerät beispielsweise alle vorhandenen Hardware-Schnittstellen auslesen und somit entscheiden, inwiefern diese Schwachstellen für einen sicheren Umgang mit dem Endgerät darstellen können. Im Test kam die Version 4.0.0.133 des *Trusted Mobility Servers* zum Einsatz.

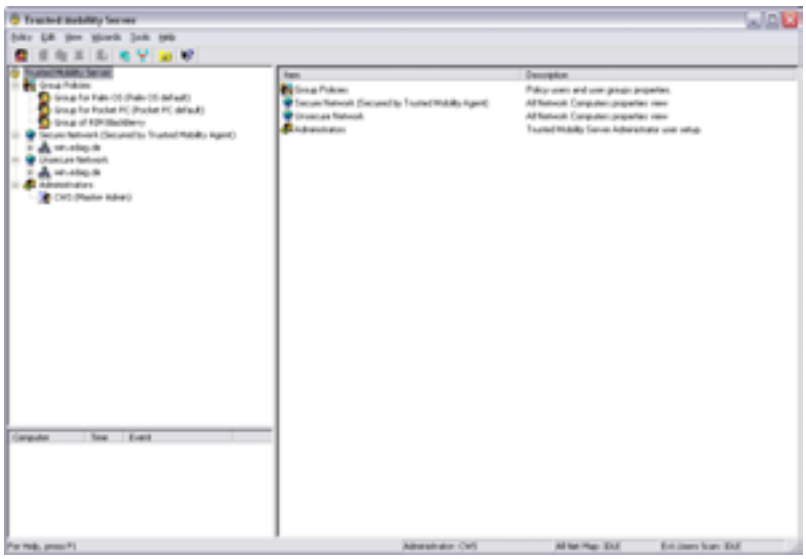


Abb. 5.9: *Trusted Mobility Server*

Die Installation verläuft wieder nach klaren Anweisungen und ist innerhalb kürzester Zeit abgeschlossen. Nach einem Neustart von *Microsoft Windows* steht der Server mit allen Funktionen zur Verfügung. Mit *Trusted Mobility Server* können Sicherheitseinstellungen für Endgeräte mit *Palm OS*, *Windows CE* und *Symbian OS* definiert werden. Die Einstellungen werden hier über *Group Policies* abgebildet, die jeweils für ein mobiles System gelten. Dies liegt wieder an den unterschiedlichen Architekturen der mobilen Betriebssysteme. Wahlweise können die Grundeinstellungen manuell oder über einen Assistenten, auch *Wizard* genannt, erstellt werden. Der *Wizard* ist sehr gut struktuiert und fragt alle möglichen Einstellungen ab. Hier ist die Menge der Optionen sogar umfangreicher als bei der *Pointsec*-Lösung. Es können Passwörter für das komplette Endgerät, aber auch für einzelne Anwendungen definiert werden und es gibt die Möglichkeit eines Administrator-Passworts. Mit diesem Passwort kann im Notfall zusätzlich auf das Endgerät zugegriffen werden. Als Passwort können hier neben einem alphanumerischen Passwort oder einer *PIN* Bilder zur Authentifizierung verwendet werden. Bei *Palm OS* können sogar die Hardware-Tasten des Gerätes als Passwort verwendet werden. In der *Group Policy* können genaue Restriktionen für die Verwendung von eingebauter Hardware erstellt werden. Die Verwendung einer eingebauten Digitalkamera

oder von *WLAN*- bzw. *Bluetooth* kann darüber komplett unterbunden werden. Auch die Zeit in der das Endgerät benutzt werden darf, z.B. nur während der Bürozeiten, ist einstellbar. Ebenfalls kann definiert werden, ob der Benutzer eigene Software installieren darf oder nicht. Eine weitere Einstellung erlaubt die Definition, welche Dateien unbedingt auch nach einem Hardware-Reset auf dem Endgerät sein müssen. Zum Schluß bietet der *Trusted Mobility Server* die Auswahl der Verschlüsselung der abgelegten Daten. Zur Auswahl stehen die Verschlüsselungsalgorithmen *AES* mit einer maximalen Schlüssellänge von 256 Bit und *3DES*. Jetzt fehlt nur noch die Einstellung welche Verzeichnisse bzw. welche Daten verschlüsselt werden sollen und die Definitionen von Standardwerten ist abgeschlossen.

Damit der Benutzer gar nicht mitbekommt, dass die Client-Software *PDASecure* auf dem Endgerät läuft, kann in der *Group Policy* definiert werden, dass die Software vor dem Benutzer versteckt wird. Einzig und alleine die Anmeldung an dem mobilen Endgerät wird sich dann für den Benutzer ändern. Somit können alle individuellen Einstellungen des Benutzers untersagt werden. Damit die *Group Policy* die *PDASecure*-Software auf dem mobilen Endgerät installiert wird, erstellt man im *Trusted Mobility Server* ein *Installation Set*, dass dann über eine Synchronisationslösung, wie z.B. *Pylon Anywhere*, auf die Endgeräte verteilt wird. Der *Trusted Mobility Server* läßt sich auch hier in unterschiedliche Synchronisationslösungen integrieren, vor allem aber in die beiden oben beschriebenen Lösungen. Aber auch wieder nur, um die *Installation Sets* oder geänderte *Group Policies* zu verteilen. Die Installation auf dem mobilen Endgerät läuft auch hier ohne Probleme ab und als Benutzer merkt man nicht, dass hier zusätzliche Sicherheitsfunktionen aktiv sind. Leider hatten alle mobilen Testgeräte keine Unterstützung für *WLAN* oder *Bluetooth*, geschweige denn eine eingebaute Digitalkamera. Deswegen konnte ich diese Funktionen nicht selbst testen. Bei meinem Besuch der *SYSTEMS 2004* in München habe ich diese Einschränkungen in einer Live-Demonstration aber in Aktion gesehen und sie haben vollkommen überzeugt. Die Möglichkeit mit der Bilderkombination als Passwort-Eingabe hat sich auch hier im praktischen Einsatz als sehr hilfreich und schnelle Lösung erwiesen.

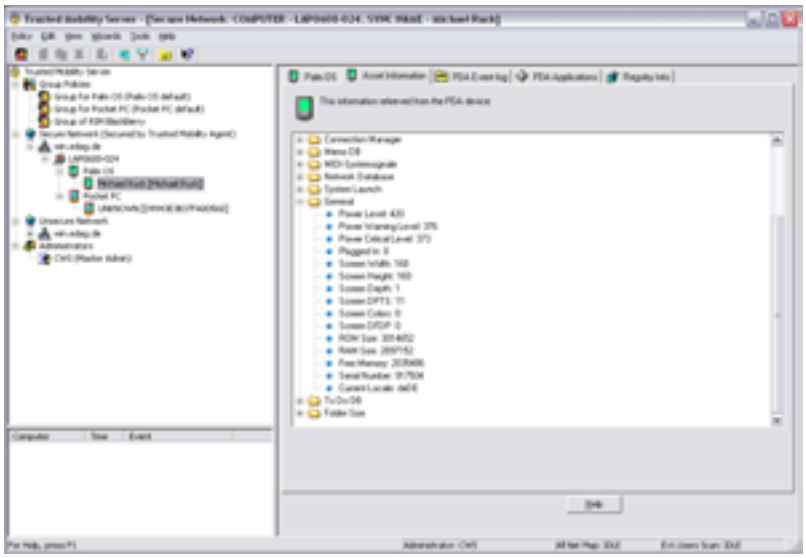


Abb. 5.10: *Informationsmanagement im Trusted Mobility Server*

Der *Trusted Mobility Server* bietet neben dem Sicherheitsmanagement wie oben beschrieben auch ein Informationsmanagement über die unterschiedlichen Endgeräte. Und das gilt nicht nur für Laptops oder *PDAs*, sondern für alle Rechner in einem Netzwerk. Voraussetzung ist die Installation des *Trusted Mobility Agents* auf einem Laptop oder Desktop-PC. Der *Trusted Mobility Server* durchsucht dann das Netzwerk nach diesem *Agent* und listet anschließend alle gefundenen Systeme in einer Baumstruktur auf. Dient ein System zusätzlich als Host-System für einen *PDA* oder ein *Smartphone*, so werden diese Endgeräte als Unterpunkt am angeschlossenen Host-System angezeigt. Somit hat man sehr schnell eine Übersicht, welches mobile Endgerät an welchem Host-System angeschlossen ist und kann darüber auch Rückschlüsse ziehen, wer das Engerät momentan benutzt. Der *Trusted Mobility Agent* liefert eine ganze Menge an Informationen über das System auf dem er installiert ist und die evt. daran angeschlossenen mobilen Endgeräte. Die Liste der Informationen reicht von den installierten Anwendungen, bis hin zu den vorhandenen Hardware-Schnittstellen. Auch Informationen aus Log-Dateien können ausgelesen werden, um evt. bei Problemen nach Fehlern zu suchen. Dieses Informationsmanagement dient ebenfalls dazu evt. Schwachstellen zu erkennen, wenn ein Gerät beispielsweise eine Hardware-Schnittstelle unterstützt, die über eine *Group Policy* noch nicht vollständig geschützt wird. Leider können die hier angezeigten Informationen nicht exportiert werden, um sie evt. in anderen Anwendungen zu verwenden. Hier ist die Auswertungsfunktion des *Pylon Anywhere* sinnvoller, vor allem weil hier auch nur die Informationen der mobilen Endgeräte angezeigt werden. Informationen über Laptops oder Desktop-PCs werden bereits in der Systemverwaltung des *EDAG*-Netzwerkes gesammelt und ausgewertet.

Für die Zukunft ist geplant, auch die Verschlüsselung von PCs mit der *TRUST Enterprise Secure*-Lösung zu ermöglichen, was heute noch komplett fehlt. Dabei wird das gleiche Verschlüsselungsframework zum Einsatz kommen, das auch schon in der *PDASecure*-Software verwendet wird. Dieses Framework ist nach dem amerikanischen Sicherheitsstandard *FIPS 140-2* der *NIST* zertifiziert.

Die Anschaffungskosten teilen sich hier der *Trusted Mobility Server* und den Lizenzen für *PDASecure*. Der Server alleine schlägt mit 7.500,- € zu Buche. *PDASecure* ist als *Gerätelizenz* erhält mit ca. 69,- € pro Lizenz. Wie schon bei den anderen getesteten Lösungen betragen die laufenden Kosten für Software Updates und Uprades, sowie für Support per eMail oder Telefon jährlich 20% der Anschaffungskosten. Für die Installation des *Trusted Mobility Servers* ist mit 2 Manntagen zu rechnen, wobei diese mit externer Unterstützung ablaufen sollte. Während dieser Zeit wird der *Trusted Mobility Server* in die bestehende Netzwerkstruktur eingebunden und die Administratoren des Servers geschult. Nach der Installation und der Definition der *Group Policies* ist an dem Server auch nicht mehr viel zu machen. Für neue mobile Endgeräte muss nur sicher gestellt werden, dass die *PDASecure*-Software automatisch mit der dazugehörigen *Policy* installiert wird. Das kann aber durch die Integration in die *Pylon Anywhere*-Lösung realisiert werden.

Alle Bewertungen sind noch einmal in der Tabelle 5.4 zusammengefasst.

Allgemeine Kriterien

	1	2	3	4	5
einheitliche Administrierung		×			
Administrierung mobiler Systeme	×				
Einstellungen für Benutzer vordefinierbar	×				
einfache Handhabung für Administratoren und Benutzer	×				
Kopplung mit Benutzerverwaltung	×				
Integrierbarkeit in Netzwerk	×				
Kombinierbar (optional)		×			

Spezielle Kriterien für Sicherheitslösung

	1	2	3	4	5
Aktuelle Sicherheitsstandards	×				
Zwei-Faktor-Authentifizierung (optional)			×		
Detaillierte Sicherheitseinstellungen (optional)	×				

Kriterien für Kosten

	1	2	3	4	5
Anschaffungskosten			×		
Support-Kosten			×		
personelle Kosten		×			

Tabelle 5.4: *Bewertungsbogen Trusted Mobility Server*

5.3 Fazit

Abschließend ist festzustellen, dass es viele unterschiedliche Lösungen gibt, die zum einen für die Daten-Synchronisation, zum anderen für die Sicherheit der mobilen Endgeräte gut geeignet sind. Es fehlt aber eine Lösung, die beide Bereiche miteinander vereint und zufriedenstellend abdeckt. Weiterhin exisitieren nur wenige Lösungen, die möglichst viele unterschiedliche mobile Systeme unterstützen. Sicherheitslösungen sind oft nur für *Windows CE* oder nur für *Palm OS* geeignet und bieten keine zentrale Administrierung. Genauso sieht es bei Lösungen für die Daten-Synchronisation aus.

Die von mir getesteten Lösungen sind aber gute Beispiele, dass es auch anders gehen kann und stellen die heutigen Möglichkeiten für eine All-In-One-Lösung dar. Aufgrund der großen Unterschiede die teilweise zwischen den einzelnen mobilen Systemen bestehen, wird sich so schnell auch keine Lösung finden, die über eine *Policy* gleichtzeitig alle Systeme abdeckt. Auch eine Komplettlösung, die sowohl die Möglichkeit der Daten-Synchronisation mit einer Groupware, als auch genügend Sicherheitseinstellungen für ein mobiles Endgerät bietet, ist momentan nicht zu finden. Somit bleibt nur der Weg für beide Bereiche eine geeignete Lösung auszuwählen. Es bleibt zwar die Trennung der Administrationsoberflächen, aber zumindest bei der Datenverteilung arbeiten beide getesteten Sicherheitslösungen mit beiden getesteten Synchronisationslösungen sehr gut zusammen. Zu beachten sind aber auf alle Fälle noch die hohen Anschaffungskosten von ca. 200.000,- € allein um die bestehenden Geräte bei der *EDAG Engineering & Design AG* mit den vorgestellten Lösungen auszustatten.

Allgemein ist zu sagen, dass alle getesteten Lösungen aber durchwegs überzeugen. Die Installation ist einfach durchzuführen und die Integration in das bestehende Netzwerk der *EDAG Engineering & Design AG* gestaltet sich auch problemlos. Allerdings muss noch die direkte Synchronisation eines mobilen Endgerätes von extern gestestet werden. Die zentrale Administrierung ist einfach und verständlich gehalten und eine Einarbeitung ist in kurzer Zeit möglich. Dass die Administrierung und die Verteilung von Software und Daten zentral geschieht ist auch alleine schon ein großer Vorteil dieser Lösungen. Auch die Benutzer profitieren von diesen Lösungen. Die Bedienung vereinfacht sich, weil alles über eine Software abgebildet wird, die auf jedem System gleich ist. Darüber hinaus werden die Funktionalitäten, z.B. durch die Sicherheitslösung, des mobilen Endgerätes erheblich erweitert.

Als Lösung für den Bereich „Datenkommunikation, Regeln für Benutzer und Inventarisierung“ empfehle ich die *Pylon Anywhere*-Lösung von *Sybase*. Die Einstellungen sind umfangreicher als bei der *OneBridge*-Lösung von *Extended Systems* und Auswahlmöglichkeiten der Verbindungsverschlüsselung tragen zu mehr Sicherheit schon bei der Datensynchronisation bei. Hier bietet der *OneBridge*-Server nur *SSL* als Möglichkeit, wobei *Pylon Anywhere* auch mit *AES* die übermittelten Daten verschlüsselt. Ebenso ist das *System Management* mit seinen Reporting- und Inventarisierungsfunktionen ein Punkt, der die *Pylon Anywhere*-Lösung interessanter macht. Diese Möglichkeit fehlt bei der *OneBridge*-Lösung komplett. Bei diesem Umfang kann man auch verschmerzen, dass es bei *Pylon Anywhere* keinen einheitlichen *Desktop Connector* gibt und man weiterhin auf *ActiveSync* und/oder *HotSync* angewiesen ist. Letzter Grund ist auch der günstigere Preis gegenüber der *OneBridge*-Lösung. D.h. man bekommt für weniger Geld mehr an Funktionen und an Leistung. Auch dass der Vertrieb der Software über ein Unternehmen aus Fulda geschieht, kann im Support-Fall von Vorteil sein. Eine genaue Evaluierung dieser Lösung ist dringend zu empfehlen.

Die Auswahl einer Sicherheitslösung gestaltet sich schwieriger. *Pointsec* ist Weltmarktführer mit seinen Lösungen und bietet schon jetzt eine Verschlüsselung auch für PCs an. Außerdem sind einige Produkte von *Pointsec* über einen internationales Sicherheitsstandard zertifiziert, was sich evt. bei der Zertifizierung der *EDAG Engineering & Design AG* nach *BS 7799*[10] positiv auswirken kann. Jedoch bietet der *Trust Mobility Server* von *Trust Digital* mehr Einstellungen und Restriktionen für die Sicherheit des Gerätes und die Benutzung der Hardware-Schnittstellen. Ähnliches plant *Pointsec* im nächsten Jahr auch für seine Software umzusetzen. In wieweit das realisiert wird und dann auch brauchbar ist bleibt abzuwarten. Aus diesem Grund empfehle ich derzeit die *Trusted Mobility Server*-Lösung, da dem Benutzer hier eindeutig mehr Vorgaben gemacht werden können, als es bei den *Pointsec*-Produkten der Fall ist. Dennoch sollte vor der endgültigen Auswahl einer Sicherheitslösung beide Produkte nochmal genauer betrachtet werden, inwiefern sich hier bereits Veränderungen ergeben haben. Denn derzeit werden die Sicherheitslösungen für mobile Endgeräte kontinuierlich weiter entwickelt und fast wöchentlich neue Versionen auf den Markt gebracht.

[10] *siehe* Kapitel 6.1.2 auf Seite 76

Kapitel 6

Standardisierte Sicherheit

Die Bedrohung eines Firmennetzwerkes hat heute viele Ursachen. Auf der einen Seite stehen die technischen Gefahren, wie Angriffe aus dem Internet oder auch der Ausfall von Systemen, durch die ebenfalls ein wirtschaftlicher Schaden entstehen kann. Durch Angriffe aus dem Internet können zusätzlich Daten an die Öffentlichkeit gebracht werden, die nicht für diese gedacht sind. Nicht zu verachten ist ebenfalls die Bedrohung durch die eigenen Mitarbeiter. Die Motivation kann hierbei von reiner Neugier, was so alles im Netz möglich ist, über Fahrlässigkeit bis hin zu einer gezielten Sabotage reichen. In Zeiten einer schlechten wirtschaftlichen Lage kann es mitunter verstärkt vorkommen, dass frustrierte Mitarbeiter ein Unternehmen sabotieren. Hier reichen die Delikte von der Weitergabe geheimer Unternehmensdaten bis hin zur gezielten Manipulation von wichtigen IT-Systemen. Nach einer Studie vom *Bundesamt für Sicherheit in der Informationstechnik* und der Unternehmensberatung *Mummert Consulting* aus dem Jahr 2003 ist jeder vierte Angriff auf ein Unternehmensnetzwerk durch verärgerte oder frustrierte Mitarbeiter zurückzuführen[1].

Vor solchen Attacken kann man sich als Unternehmen nur schützen, wenn entsprechende Regeln für die Mitarbeiter definiert und auch eingehalten werden. Durch Kunden kann zusätzlich die Anforderung gestellt werden, die zur Verfügung gestellten Daten und Informationen vor einem unerlaubten Zugriff zu schützen. Damit nicht jedes Unternehmen seine eigenen Regeln aufstellt, die evt. nicht nachvollziehbar sind, gibt es die Möglichkeit seine Sicherheitspolitik nach standardisierten Verfahren aufzubauen. Dies dient zur Transparenz der Sicherheitspolitik gegenüber dem Kunden und zu einer einheitlichen Regelung beim vertrauensvollen Umgang mit Daten und Informationen. Aber auch unabhängig von bestehenden Kunden wird die Zertifizierung nach einem anerkannten Standard für die Zukunft immer wichtiger. Durch spezielle Anforderungen in der Automobilindustrie, der auch die *EDAG Engineering & Design AG* angehört, oder die Vereinbarungen der Finanzdienstleister u.a. über Regelungen bei der Kreditvergabe (*Basel II*), ist eine standardisierte Sicherheitspolitik unumgänglich. Denn nur durch standardisierte Verfahren ist eine Vergleichbarkeit der Sicherheitspolitik verschiedener Unternehmen gegeben und für eine sachliche Bewertung geeignet.

[1] *siehe* `http://www.heute.t-online.de/ZDFheute/artikel/25/0,1367,COMP-0-2036601,00.html`

In der Regel gliedert sich ein Sicherheitsstandard in einen *technischen*, einen *organisatorischen* und einen *rechtlichen* Teil. Der *technische* Teil umfasst zum einen den Schutz von IT-Systemen, aber auch die technische Realisierung von Zugangskontrollen, die meist ja mit IT-Systemen gekoppelt sind. Im Detail wird hier beispielsweise definiert, dass für den Schutz des Unternehmsnetzwerkes gegenüber dem Internet Firewalls und *DMZ*[2]-Umgebungen eingesetzt oder auf Servern, Desktop-PCs und Laptops Virenscanner installiert werden. Im *organisatorischen* Teil werden Regeln für die Mitarbeiter aufgestellt, wie sie z.B. mit vertraulichen Informationen und Unternehmensdaten umgehen müssen oder was bei der Erkennung einer direkten Bedrohung des Unternehmens zu tun ist. Dieser Teil ist meiner Meinung nach das wichtigste Element bei einer Zertifizierung. Kaum eine Bedrohung ist heute größer als der eigentliche Benutzer selbst, egal ob dieser die Bedrohung bewußt oder unbewußt auslöst. Deshalb ist die Definition von klaren Regeln sehr wichtig und sollte mit größter Sorgfalt erarbeitet werden. Allgemein müssen im *organisatorischen* Teil alle Maßnahmen und Regeln definiert werden, die technisch nicht realisiert werden können. Als letzter Teil sollte der *rechtliche* Aspekt nicht außer Acht gelassen werden. Bei der Aufstellung einer umfassenden Sicherheitspolitik dürfen Gesetze nicht verletzt und müssen rechtliche Vorgaben jederzeit eingehalten werden. Wichtig hier in Deutschland sind hier u.a. das *Bürgerliche Gesetzbuch* (*BGB*) und damit verbunden das *Grundgesetz* (*GG*), aber auch das *Bundesdatenschutzgesetz* (*BDSG*) und viele andere Gesetze. Ebenfalls sind neue Gesetze, wie z.B. dem *Gesetz zur Kontrolle und Transparenz im Unternehmensbereich* (*KonTraG*), das die Unternehmensleitung dazu verpflichtet, bedrohliche Schwachstellen aufzudecken und Maßnahmen gegen diese Schwachstellen zu definieren, relevant für eine rechtliche Betrachtung. Andernfalls ist die Unternehmensleitung für die, durch die Schwachstellen entstandenen Schäden direkt haftbar[3]. Somit steht der *rechtliche* und der *organisatorische* Teil einer Sicherheitspolitik in einem direkten Zusammenhang. In meiner weiteren Betrachtung beziehe ich mich jedoch hauptsächlich auf die *technischen* und *organisatorischen* Teile der Sicherheitspolitik.

6.1 Zertifikate für Sicherheit

Viele internationale und nationale Standards stehen heute zur Verfügung, durch die eine Zertifizierung nach unterschiedlichen Kriterien möglich ist. Dabei reicht die Bewertung von einem allgemeinen Sicherheitskonzept mit Gebäudeschutz usw. bis hin speziell zur Sicherheit in der IT. Drei sehr wichtige Standards werden im Folgenden etwas näher betrachtet.

6.1.1 Common Criteria

Die *Common Criteria for Information Technology Security Evaluation*, kurz *CC*, sind der Zusammenschluß und Weiterentwicklung der europäischen *Information Technology Security Evaluation Criteria* (*ITSEC*), des amerikanischen *Orange Book* und den *Canadian Trusted Computer Evaluation Criteria* (*CTC-*

[2]DMZ: Demilitarisierte Zone — Schnittstelle zwischen einem Unternehmensnetzwerk und einem externen Netz, wie z.B. dem Internet

[3]*siehe* `http://www.micic.com/knowledgebase/glossar/go-kontrag.html`

PEC). Seit Version 2.1 vom April 1999 sind die *CC* darüber hinaus als internationaler Standard *ISO 15408* veröffentlicht. Durch die *CC* wird ausschließlich die Sicherheit von Hard- und Software überprüft und zertifiziert.

Dazu stellen die *CC* zum einen funktionale Sicherheitsanforderungen und zum anderen Anforderungen an die Vertrauenswürdigkeit. Daraus resultieren Schutzprofile und Sicherheitsvorgaben, die dann für das *Target of Evaluation* gilt. Das *Target of Evaluation* ist dabei ein IT-System, z.B. ein Unternehmensnetzwerk oder eine einzelne Netzwerkkomponente, wie beispielsweise ein Router oder ein *Appliance Server*[4], inkl. aller Administratoren, Hard- und Software und Dokumentationen. Der Katalog an funktionalen Sicherheitsanforderungen bietet eine Übersicht der Zusammenhänge zwischen Bedrohungen und daraus resultierenden Sicherheitsanforderungen. Die Beschreibungen werden dazu in elf Klassen unterteilt, die jeweils einen Teil des Sicherheitskonzeptes darstellen. Die Klasse *FDP* beschreibt beispielsweise den *Schutz der Benutzerdaten*, also Einzelheiten zur *Password policy* oder die Kontrolle über den Informationsfluss. Die Einhaltung dieser Beschreibungen wird dringend empfohlen, es kann in begründeten Fällen aber auch davon abgewichen werden. Wann dies der Fall ist entscheidet der Auditor. Die Anforderungen an die Vertrauenswürdigkeit geben an, in welchem Grad man einem geprüften IT-System vertrauen kann. Dazu existieren folgende sieben *Evaluation Assurance Level* (*EAL*):

EAL-1	funktionell getestet
EAL-2	strukturell getestet
EAL-3	methodisch getestet und überprüft
EAL-4	methodisch entwickelt, getestet und durchgesehen
EAL-5	semiformal entworfen und getestet
EAL-6	semiformal verifizierter Entwurf und getestet
EAL-7	formal verifizierter Entwurf und getestet

Im Standard *ISO 15408* wird der *EAL* bis Level 4 international anerkannt. Die Anforderungen an die Vertrauenswürdigkeit ist ebenfalls in Klassen eingeteilt, die jeweils einen Teil der Evaluierung repräsentieren. So gibt es beispielsweise die Klasse *Handbücher* (*AGD*), in der Kriterien über die Qualität der Handbücher festgelegt wurden, oder die Klasse *Lebenszyklus-Unterstützung* (*ALC*), die Vorgaben für die Fehlerunterstützung und die Lebenszeit des geprüften IT-Systems macht. Zu jeder Klasse gehören Schutzprofile, durch die die Anforderungen an die Funktionalität und die Vertrauenswürdigkeit näher beschrieben werden. In diesen Schutzprofilen werden darüber hinaus evt. Bedrohungen des IT-Systems genau erläutert und den jeweiligen Anforderungen gegenüber gestellt. So entsteht ein komplettes Sicherheitskonzept für das geprüfte IT-System. Die Schutzprofile stellen dabei auch das Grundgerüst für die *Evaluation Assurance Level* dar. Dabei ist genau festgelegt, welches Vertrauenswürdigkeit-Schutzprofil erfüllt sein muss um ein bestimmtes *EAL* zu erreichen.

Die *Common Criteria* sind heute wichtig für Entwickler von Netzwerkkomponenten, die gewisse Sicherheitsfunktionen übernehmen sollen, wie beispielsweise *Firewall Appliances* oder *Anti-Viren Appliances*. Für die Evaluierung eines kompletten Unternehmensnetzwerkes

[4]Ein *Appliance Server* ist ein Server-System mit einer speziellen Aufgabe (z.B. Firewalling), bei dem alle dazu nötigen Funktionen bereits vorinstalliert sind. Das System hat ein direkt auf die Hardware zugeschnittenes Betriebssystem und eine dazugehörige Administrationsoberfläche. Die Systeme haben meist die 19“-Bauform. *siehe* `http://www.webopedia.com/TERM/S/server_appliance.html`

sind diese Kriterien ungeeignet, da es sich wirklich nur um eine rein technische Überprüfung von Sicherheitskriterien handelt und die Benutzer des Netzwerkes aus organisatorischer Sicht nicht berücksichtigt. Der Einsatz von technischen Geräte, die den *Common Criteria* entsprechen ist aber sicherlich von Vorteil. Sie bieten eine überprüfbare Sicherheit, die der Hersteller durch seine Zertifizierung garantiert. Vor allem beim Aufbau eines Sicherheitskonzeptes für das komplette Unternehmen und der Zertifizierung nach einem anerkannten Sicherheitsstandard werden IT-Systeme mit *CC*-Zertifikat von einem Auditor als positiv gewertet. *CC*-zertifizierte Geräte erreichen heute maximal *EAL-4*, was durchaus ausreichend ist und dem internationalen Standard *ISO 15408* entspricht. Das *Bundesamt für Sicherheit in der Informationstechnik* (*BSI*) war an der Entwicklung der *CC* Version 2.1 und dem Standard *ISO 15408* beteiligt und hat eine deutsche Version der *Common Criteria* auf seiner Homepage unter [4–6] veröffentlicht.

6.1.2 BS 7799 / ISO 17799

Für die Zertifizierung eines unternehmensweiten Sicherheitskonzeptes eignet sich hierfür besser der *British Standard 7799* (*BS 7799*), das dem internationalen Standard *ISO 17799*[5] entspricht. Allgemein kann man das *BS 7799* als ein Konzept für ein *Informationsschutzmanagement System* (*ISMS*) bezeichnen, in dem sowohl der Schutz von Gebäuden und Einrichtungen, aber auch der kompletten IT beschrieben ist. Somit deckt das *BS 7799* sowohl den technischen, als auch den organisatorischen Schutz von Unternehmensinformationen ab, immer mit dem Augenmerk auf die Einhaltung rechtlicher Aspekte. In der ersten Version des Standards (*BS 7799:1999*), die 1999 veröffentlicht wurde, wird nur das Erstellen eines solchen *ISMS* gefordert. In der aktuellen Version aus dem Jahr 2002 (*BS 7799-2:2002*) wird darüber hinaus auch die Pflege, die Aktualität, die Handhabung und die Einhaltung des *ISMS* gefordert. Die *EDAG Engineering & Design AG* ist seit September 2004 nach der Version *BS 7799-2:2002* mit dem Hauptsitz in Fulda und der Niederlassung in München zertifiziert.

Im Detail wird bei der *BS 7799* zwischen einer allgemeinen Sicherheitspolitik, organisatorischen Maßnahmen für Sicherheit, Klassifizierung und Kontrolle von Lagerbeständen, Sicherheitsinstruktionen für Mitarbeiter, physikalische Sicherheitsfaktoren, sichere Verwaltung der Kommunikationseinrichtungen und Tätigkeiten der Mitarbeiter, Zugangskotrollen zu den Unternehmensinformationen, Sicherheit bei der Systementwicklung und -betreuung, Notfallpläne und die rechtlichen Einhaltung unterschieden. Es ergibt sich also neben einem reinen Sicherheitskonzept auch Ansätze für eine Qualitätssicherung[6]. In jedem dieser Bereiche sind Maßnahmen definiert, die als Grundlage für die Erstellung einer Sicherheitspolitik für das gesamte Unternehmen verwendet werden können. Beispielsweise wird genau definiert, welche Vorkehrungen in einem Gebäude getroffen werden müssen, um es vor evt. Gefahren zu schützen. Ganz einfaches Beispiel wäre hier das Aufhängen von Feuerlöschern, die im Fall eines Brandes leicht erreichbar sind. Bei der Zertifizierung wird dazu eine Dokumentation angefertigt, die zu jedem dieser benötigten Maßnahmen klar definiert, wie diese im Unternehmen umgesetzt werden. Diese Dokumentation ist damit die schriftliche Fixierung

[5] Entspricht der Version *BS 7799:2000* und nicht der aktuellen *BS 7799-2:2002*

[6] Ein Vergleich mit der *ISO 9000* findet man in [3] ab Seite 28

der kompletten Sicherheitspolitik des Unternehmens.

Die Zertifizierung nach *BS 7799* gewinnt immer mehr an Bedeutung und hat zukünftig einen ähnlichen Stellenwert wie der *ISO 9000*-Standard für die Qualitätssicherung. Bei den wachsenden Gefahren auf ein Unternehmen und insbesondere der verwendeten IT-Systeme wird eine standardisierte Sicherheitspolitik gefordert, bevor geschäftliche Beziehungen aufgebaut bzw. weiter geführt werden. Ähnlich ist es bei der *EDAG Engineering & Design AG* geschehen, die aufgrund der Anforderung eines Kunden die Zertifizierung nach *BS 7799* durchgeführt hat.

6.1.3 BSI-Grundschutz

Das vom *BSI* vergebene *BSI IT-Grundschutz*-Zertifikat ist in Deutschland ein wichtiger nationaler Standard für Sicherheit im Unternehmen. Die Anforderungen des *BSI IT-Grundschutz* entspricht hierbei weitesgehend denen des *BS 7799*[7]. Das *BSI* hat jedoch ein umfangreiches Handbuch für eine Sicherheitspolitik erstellt, das ca. 2.500 Seiten umfassende *Grundschutzhandbuch* [8]. In diesem Handbuch werden alle Gefahren und daraus resultierende Maßnahmen viel ausführlicher definiert, als es bei *BS 7799* der Fall ist. Die Gefahrenstufen werden in die Bereiche *Höhere Gewalt*, *Organisatorische Mängel*, *Menschliche Fehlhandlungen*, *Technisches Versagen* und *Vorsätzliche Handlungen* eingeteilt, die Schutzmaßnahmen in die Bereiche *Infrastruktur*, *Organisation*, *Personal*, *Hardware/Software*, *Kommunikation* und *Notfallsorge* untergliedert. Zusätzlich werden die Gefahren und die Maßnahmen einer Dringlichkeitskategorie (A = sehr dringend, B = mittel dringend, C = bei Bedarf) zugeordnet bzw. als komplett als „optional“ definiert.

Für den Aufbau eines umfassenden Sicherheitskonzeptes werden im *Grundschutzhandbuch* verschiedene Hauptkomponenten definiert, denen die dazugehörige Gefahrenstufe und die daraus resultierenden Schutzmaßnahmen zugeordnet sind. Diese Komponenten unterteilen sich in *IT-Grundschutz übergeordneter Komponenten*, *IT-Grundschutz im Bereich Infrastruktur*, *Nicht vernetzte Systeme und Clients*, *Vernetzte Systeme*, *Datenübertragungseinrichtungen*, *Telekommunikation* und *Sonstige IT-Komponente*. In der Komponente *IT-Grundschutz übergeordneter Komponenten* existiert beispielsweise eine Unterkomponente *3.2 Personal* der nun u.a. die Gefahrenlagen *G 1.1 Personalausfall*, *G 2.2 Unzureichende Kenntnis über Regelungen*, *G 3.3 Nichtbeachtung von IT-Sicherheitsmaßnahmen* und *G 5.104 Ausspähen von Informationen* zugeordnet sind. Darauf ist u.a. durch die Maßnahmen *M 3.1 Geregelte Einarbeitung/Einweisung neuer Mitarbeiter*, *M 3.2 Verpflichtung der Mitarbeiter auf Einhaltung einschlägiger Gesetze, Vorschriften und Regelungen* und *M 3.5 Schulung zu IT-Sicherheitsmaßnahmen* zu reagieren[8]. Insgesamt existieren 60 Komponenten, auf die 333 Gefahrensituationen und 772 Schutzmaßnahmen mehrfach verteilt werden. Es kann somit sein, dass eine Gefahrensituation und Schutzmaßnahme für mehrere Komponenten gilt.

Das *BSI* bietet nebenbei viele Hilfsmittel, um das *Grundschutzhandbuch* im Unternehmensumfeld umzusetzen. Die Hilfsmittel reichen hier von Muster von Merkblättern, Fra-

[7] Eine Gegenüberstellen von *BS 7799* und dem *BSI IT-Grundschutz* findet man unter `http://www.bsi.bund.de/gshb/deutsch/aktuell/bs7799.htm`

[8] Detaillierte Aufstellung in [8] auf Seite 87

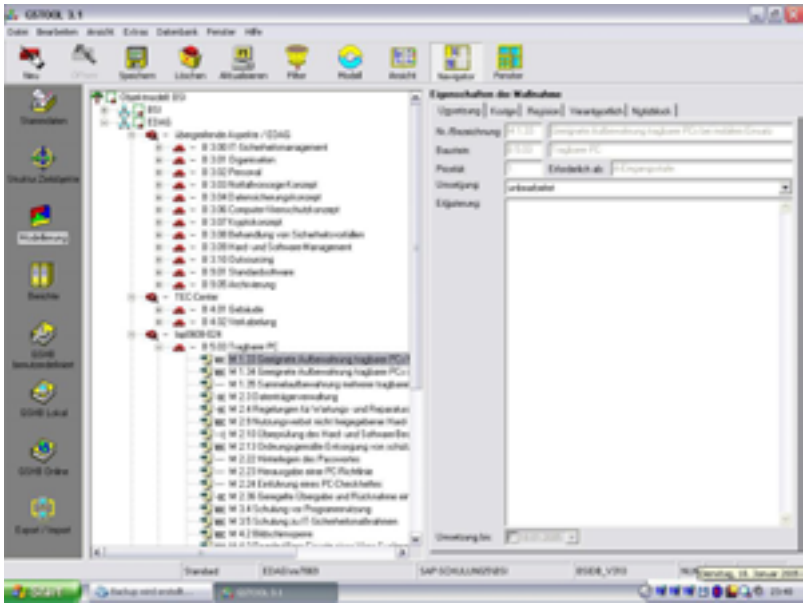


Abb. 6.1: *GSTOOL des BSI*

gebögen oder Betriebsvereinbarungen zum Thema *IT-Sicherheit* bis hin zu kompletten IT-Sicherheitsrichtlinien und -konzepten[9]. Als besonderes Hilfsmittel bietet das *BSI* das *IT-Grundschutz-Tool* (*GSTOOL*). Dieses Tool bietet eine Visualisierung des *Grundschutzhandbuches*, indem man die komplette Infrastruktur eines Unternehmens, mit allen Gebäuden, Abteilungen, Mitarbeiter und IT-Systemen abbilden kann. Dadurch entsteht ein direkt auf das Unternehmen zugeschnittenes Paket von Gefahrensituationen und damit verbundenen Schutzmaßnahmen. Des weiteren können zu jeder eingepflegten Komponente verantwortliche Personen und die dazugehörigen Kosten für die Umsetzung und Erhaltung der Schutzmaßnahme definiert werden. Über verschiedene Reporting-Funktionen erhält schnell eine Übersicht der Kosten für die Sicherheit im Unternehmen, was vor allem für die Unternehmensführung interessant ist. Das *GSTOOL* erstellt ebenfalls Listen der noch nicht umgesetzten Sicherheitsmaßnahmen, sortiert nach den Dringlichkeitskategorien. So kann man die weitere Umsetzung des Sicherheitskonzeptes besser koordinieren. Um ein komplettes Unternehmen, vor allem wenn es weltweit Standorte und mehrere IT-Systeme im Einsatz hat, im *GSTOOL* abzubilden, Bedarf es jedoch einen gewissen Aufwand. Hat man aber alles abgebildet, so bietet das Tool eine gute Dokumentation des Sicherheitskonzeptes und einen guten Überblick über mögliche Gefahrenstellen, die noch nicht behoben sind. Nebenbei hat man zusätzlich noch wesentliche Teile des *BS 7799* umgesetzt, wenn alle Schutzmaßnahmen umgesetzt wurden, die das *GSTOOL* vorschlägt. Das *GSTOOL* ist direkt über das *BSI* zu beziehen und kann von dessen Webseite in einer 30-Tage-Testversion heruntergeladen werden. Der Lizenzpreis für die Vollversion liegt bei 765,- € pro Einzelplatzlizenz (zzgl. MwSt).

[9]komplette Liste der Hilfsmittel: *siehe* `http://www.bsi.bund.de/gshb/deutsch/hilfmi/hilfmi.htm`

6.2 Sicherheitspolitik bei der *EDAG*

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

6.2.1 Unternehmensweite Sicherheitspolitik

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

6.2.2 Sicherheit in der IT

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

6.2.3 Stand der Zertifizierung nach BS 7799

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

6.3 Mobile Endgeräte in Sicherheitskonzepten

Viele Sicherheitskonzepte betrachten oftmals nur die allgemeine IT-Netzstruktur bis hin zu Desktop-PCs und Laptops — als einziges mobile Endgerät. Die wachsende Gefahr von kleineren mobilen Endgeräte, wie PDAs, Mobiltelefone und SmartPhones, wird dabei oft außer Acht gelassen. Wie in den vorherigen Kapiteln beschrieben haben eben diese Geräte heute eine Leistungsfähigkeit und auch Speicherkapazität, was sie für eine Sicherheitsbetrachtung auf jeden Fall relevant machen.

Schon allein die Größe der Geräte machen sie zu einer Gefahr. Leicht sind die Geräte in einer Jackentasche in ein Unternehmen einzuschleusen und können somit zur Datenspionage benutzt werden. Die eingebauten Speichererweiterungen fassen heute mehrere Gigabyte und können somit auch größere Daten transportieren, was vor wenigen Jahren noch undenkbar war. Verstärkt wird das durch mobile Speicherkarten, die ebenfalls in mobilen Endgeräten verwendet werden können. Funktechnologien wie *Bluetooth* oder *Wireless LAN* bieten hier ebenfalls einen unsichtbaren Zugang zum Firmennetzwerk, auch wenn kein offizielles *Wireless LAN* installiert ist. Meist genügt schon ein an das Netzwerk angeschlossener Laptop mit eingebauter und aktivierter *WLAN*-Karte.

Einzig die in Mobiltelefone und PDAs eingebaute Digitalkamera wird in bestehende Sicherheitskonzepte übernommen. Dies ist meist auch nur die Erweiterung eines sowieso bestehenden Kameraverbot auf dem Firmengelände. Genau das ist beispielsweise bei der *EDAG Engineering & Design AG* geschehen. Für den Schutz durch die anderen Gefahren der mobilen Endgeräte existieren kaum Regelungen in Unternehmen. Ausgenommen sind hier wieder Laptops, die mit Desktop-PCs vergleichbar sind.

In den hier beschriebenen Standards sind ebenfalls keine klaren Regelungen für diese Endgeräte zu finden. Weder in den Ausführungen zum *BS 7799-2:2002*-Standard, noch in dem sehr umfangreichen *Grundschutzhandbuch*[10] des *BSI* findet keine Gefahrenseinschätzung mit daraus resultierende Schutzmaßnahmen statt. Möchte man ein mobiles Endgerät vollständig in ein bestehendes Sicherheitskonzept integrieren, so muss man diese Einschätzungen selbst vornehmen. Hilfe bietet dafür auf alle Fälle das *Grundschutzhandbuch* des *BSI*. Die Gefährdungslage eines PDA mit *Windows CE* als Betriebssystem, *WLAN*- und *Bluetooth*-Unterstützung, sowie eingebauter Digitalkamera und der Verwendungsmöglichkeit von mobilen Speicherkarten kann wie folgt definiert werden.

Gefährdungskatalog *G 1 Höhere Gewalt*

- *G 1.1* Personalausfall
- *G 1.2* Ausfall des IT-Systems
- *G 1.4* Feuer
- *G 1.5* Wasser
- *G 1.7* Unzulässige Temperatur und Luftfeuchte

[10] Stand des *Grundschutzhandbuches* zur Zeit der Diplomarbeit: Oktober 2003

- *G 1.8* Staub, Verschmutzung
- *G 1.9* Datenverlust durch starke Magnetfelder

Anmerkungen: Die Gefährdungslage *G 1.7* kommt in Betracht, wenn der PDA beispielsweise über eine Heizung abgelegt wird. Zu der Luftfeuchte kommt noch die Gefährdungslage *G 1.5* wenn der PDA beispielsweise bei Regen in einem offenen Cabrio liegen gelassen wird. *G 1.9* ist hauptsächlich für die mobilen Speicherkarten relevant.

Gefährdungskatalog *G 2 Organisatorische Mängel*

- *G 2.1* Fehlende oder unzureichende Regelungen
- *G 2.2* Unzureichende Kenntnis über Regelungen
- *G 2.4* Unzureichende Kontrolle der IT-Sicherheitsmaßnahmen
- *G 2.5* Fehlende oder unzureichende Wartung
- *G 2.8* Unkontrollierter Einsatz von Betriebsmitteln
- *G 2.9* Mangelhafte Anpassung an Veränderungen im IT-Einsatz
- *G 2.16* Ungeordneter Benutzerwechsel bei tragbaren PCs
- *G 2.36* Ungeeignete Einschränkung der Benutzerumgebung
- *G 2.37* Unkontrollierter Aufbau von Kommunikationsverbindungen
- *G 2.62* Ungeeigneter Umgang mit Sicherheitsvorfällen
- *G 2.67* Ungeeignete Verwaltung von Zugangs- und Zugriffsrechten

Anmerkungen: In der Regel fehlen einheitliche Regelungen für den allgemeinen Einsatz von PDAs. Somit kann der Benutzer diese Regelungen auch nicht kennen (*G 2.1* und *G 2.2*). *G 2.36* beschreibt gleichzeitig noch ein Problem, das nicht wirklich lösbar ist. Kaum ein mobiles Betriebssystem kennt ein Rechtemodell für die Benutzer. Der angemeldete Benutzer hat eigentlich immer Administratorrechte auf dem PDA. Daraus resultiert auch die Gefährdungslage *G 2.67*.

Gefährdungskatalog *G 3 Menschliche Fehlhandlungen*

- *G 3.1* Vertraulichkeits-/Integritätsverlust von Daten durch Fehlverhalten der IT-Benutzer
- *G 3.2* Fahrlässige Zerstörung von Gerät oder Daten
- *G 3.3* Nichtbeachtung von IT-Sicherheitsmaßnahmen
- *G 3.8* Fehlerhafte Nutzung des IT-Systems
- *G 3.9* Fehlerhafte Administrierung des IT-Systems
- *G 3.31* Unstrukturierte Datenhaltung

- *G 3.38* Konfigurations- und Bedienungsfehler
- *G 3.43* Ungeeigneter Umgang mit Passwörtern
- *G 3.44* Sorglosigkeit im Umgang mit Informationen

Anmerkungen: Die fehlerhafte Nutzung und Administrierung des PDAs hängt oft einfach davon ab, dass nicht alle Systeme gleich sind. Für ein Unternehmen kann dies nur minimiert werden, indem ein standardisiertes Gerät eingesetzt wird, dass immer die gleiche Ausstattung an Hard- und Software hat. Dadurch läßt sich auch *G 3.38* minimieren. *G 3.43* und das damit verbundene *G 3.44* ist darauf zurückzuführen, dass zumindest von Seiten des mobilen Betriebssystems, in unserem Beispiel *Windows CE*, ein Passwort nicht standardmäßig aktiviert ist. Auch die Verwendung von einfachen Passwörtern können diese Gefahrenslage hervor rufen.

Gefährdungskatalog *G 4 Technisches Versagen*

- *G 4.7* Defekte Datenträger
- *G 4.9* Ausfall der internen Stromversorgung
- *G 4.13* Verlust gespeicherter Daten
- *G 4.25* Nicht getrennte Verbindungen
- *G 4.33* Schlechte oder fehlende Authentikation

Anmerkungen: Als interne Stromversorgung dient bei einem PDA in der Regel eine Batterie, die durch eine ständige Nutzung des Endgerätes an Leistung verliert. Ist die Leistungskapazität der Batterie erschöpft und keine externe Stromversorgung verfügbar, so kann es zu *G 4.13* kommen. Hierbei sind sowohl die installierten Anwendungen, als auch gespeicherte *PIM*-Daten gemeint. *G 4.25* betrifft in unserem Beispiel die *WLAN*- und *Bluetooth*-Unterstützung. Eine bereits aufgebaute Funkverbindung, z.B. mit dem Unternehmensnetzwerk, bleibt hierbei solange bestehen, bis sie explizit abgebaut wird und bietet eine gute Möglichkeit für einen Angriff auf das verbundene Netzwerk.

Gefährdungskatalog *G 5 Vorsätzliche Handlungen*

- *G 5.1* Manipulation/Zerstörung von IT-Geräten und Zubehör
- *G 5.7* Abhören von Leitungen
- *G 5.9* Unberechtigte IT-Nutzung
- *G 5.18* Systematisches Ausprobieren von Passwörtern
- *G 5.19* Mißbrauch von Benutzerrechten
- *G 5.20* Mißbrauch von Administratorrechten
- *G 5.21* Trojanische Pferde
- *G 5.22* Diebstahl bei mobiler Nutzung des IT-Systems

- *G 5.23* Computer-Viren
- *G 5.26* Analyse des Nachrichtenflusses
- *G 5.29* Unberechtigtes Kopieren der Datenträger
- *G 5.42* Social Engineering
- *G 5.71* Vertraulichkeitsverlust schützenswerter Informationen

Anmerkungen: Auslöser für einige Gefahren aus diesem Katalog stehen in direktem Zusammenhang mit *G 5.22*. Der Dieb kann versuchen über *G 5.18* und *G 5.42* versuchen an die Informationen auf dem PDA zu kommen, wenn dieser über einen Passwortschutz verfügt. *G 5.19* und *G 5.20* können bei unserem Beispiel zusammengelegt werden, da, wie in *G 3* beschrieben, *Windows CE* über kein Rechtemodell verfügt und der Benutzer immer Administratorrechte hat. Durch die Unterstützung von Funktechnologien unseres Beispielgerätes ergeben sich *G 5.7* und *G 5.26*. Angreifer können das Abhören und die Analyse von Nachrichten über das Funknetz evt. Authentifizierungsverfahren erkennen und für einen Angriff auf das Netzwerk verwenden. *G 5.21* und *G 5.23* sind mittlerweile auch für PDAs ein Thema und können nicht mehr vollständig außer Acht gelassen werden.

Die genauen Beschreibungen der einzelnen Gefahren können aus [8] entnommen werden.

Aus den eben definierten Gefahren ergeben sich wiederum eine gewisse Anzahl von Schutzmaßnahmen, die für einen sicheren Einsatz des mobilen Endgerätes aus unserem Beispiel definiert und umgesetzt sein müssen. Werden alle folgenden Schutzmaßnahmen realisiert, so hat man sein Sicherheitskonzept für die Verwendung des Beispiel-PDA und ähnlichen Geräten erweitert.

Maßnahmenkatalog *M 1 Infrastruktur*

- *M 1.33* Geeignete Aufbewahrung tragbarer PCs bei mobilem Einsatz
- *M 1.34* Geeignete Aufbewahrung tragbarer PCs bei stationärem Einsatz
- *M 1.35* Sammelaufbewahrung mehrer tragbarer PCs

Anmerkungen: Im *Grundschutzhandbuch* ist immer nur von einem tragbaren PC die Rede. Ein PDA ist damit aber durchaus vergleichbar. Beide Systeme können problemlos auch außerhalb des Unternehmens eingesetzt werden.

Maßnahmenkatalog *M 2 Organisation*

- *M 2.9* Nutzungsverbot nicht freigegebener Software
- *M 2.11* Regelungen des Passwortgebrauchs
- *M 2.13* Ordnungsgemäße Entsorgung von schützenswerten Betriebsmitteln
- *M 2.32* Einrichtung einer eingeschränkten Benutzerumgebung

- *M 2.35* Informationsbeschaffung über Sicherheitslücken des Systems
- *M 2.36* Geregelte Übergabe und Rücknahme eines tragbaren PCs
- *M 2.45* Regelung des Datenträgeraustausches
- *M 2.62* Software-Abnahme- und -Freigabe-Verfahren
- *M 2.63* Einrichten der Zugriffsrechte
- *M 2.160* Regelungen zum Computer-Virenschutz
- *M 2.167* Sicheres Löschen von Datenträgern
- *M 2.204* Verhinderung ungesicherter Netzzugänge
- *M 2.224* Vorbeugung gegen Trojanische Pferde

Anmerkungen: Die Maßnahmen *M 2.9* und *M 2.62* sollen gewährleisten, dass nur geprüfte Software auf dem PDA verwendet wird. Der Benutzer muss daran entweder *technisch* gehindert oder *organisatorisch* darüber informiert und angewiesen werden, dass dies nicht erlaubt ist. *M 2.32* und *M 2.63* sind technisch bei unserem Beispiel-PDA nur sehr schwer zu realisieren. Dennoch sollten alle Möglichkeiten ausgeschöpft werden, um diese Maßnahmen umzusetzen. Durch *M 2.204* ist technisch festzulegen, dass beispielsweise keine ungesicherte *WLAN*-Verbindung aufgebaut wird. Zumindest die Standardmechanismen, wie *WEP* oder *WPA*, sollten für eine gesicherte Kommunikation in der Grundeinstellung aktiviert werden.

Maßnahmenkatalog *M 3 Personal*

- *M 3.4* Schulung vor Programmnutzung
- *M 3.5* Schulung zu IT-Sicherheitsmaßnahmen
- *M 3.14* Einweisung des Personals in den geregelten Ablauf eines Datenträgeraustausches
- *M 3.23* Einführung in kryptographische Grundbegriffe
- *M 3.26* Einweisung des Personals in den sicheren Umgang mit IT

Anmerkungen: Alle Maßnahmen in diesem Katalog sollen dazu dienen, dass der Benutzer ausreichend über die Benutzung des mobilen Endgerätes und die evt. damit verbundene Gefahrenlage informiert wird. Aus diesem Grund sollte der Benutzer zusätzlich ein grundlegendes Verständnis von IT-Sicherheit erlangen. Dadurch wird der Benutzer für die Notwendigkeit des Einsatzes kryptographischer Methoden sensibilisiert und akzeptiert besser beispielsweise, dass die Netzzugänge über die Funkschnittstellen nur auf gesichertem Weg ablaufen müssen.

Maßnahmenkatalog *M 4 Hardware und Software*

- *M 4.1* Passwortschutz für IT-Systeme
- *M 4.2* Bildschirmsperre

- *M 4.3* Regelmäßiger Einsatz eines Virensuchprogrammes
- *M 4.4* Geeigneter Umgang mit Laufwerken für Wechselmedien und externen Datenspeichern
- *M 4.7* Änderung voreingestellter Passwörter
- *M 4.12* Sperren von nicht benötigten Leistungsmerkmalen
- *M 4.27* Passwortschutz am tragbaren PC
- *M 4.29* Einsatz eines Verschlüsselungsproduktes für tragbare PCs
- *M 4.31* Sicherstellen der Energieversorgung im mobilen Einsatz
- *M 4.33* Einsatz eines Viren-Suchprogramms bei Datenträgeraustausch und Datenübertragung
- *M 4.38* Abschalten von nicht benötigter Leistungsmerkmale

Anmerkungen: Für den Beispiel-PDA muss die gleiche *Passwort Policy* gelten, die auch für das komplette Netzwerk gilt. Das wird durch die Maßnahmen *M 4.1*, *M 4.2*, *M 4.7* und *M 4.27* sicher gestellt. Durch die Maßnahmen *M 4.12* und *M 4.38* ist definiert, dass nicht benötigte Schnittstellen deaktiviert werden. Existiert in einem Unternehmen beispielsweise kein *WLAN*, so sollte diese Unterstützung in dem PDA ebenfalls deaktiviert werden, da keine Notwendigkeit mehr dafür besteht. Ebenso kann mit der in den Beispiel-PDA eingebauten Digitalkamera verfahren werden.

Maßnahmenkatalog *M 5 Kommunikation*

- *M 5.68* Einsatz von Verschlüsselungsverfahren zur Netzkommunikation
- *M 5.87* Vereinbarung über die Anbindung an Netze Dritte
- *M 5.88* Vereinbarung über Datenaustausch mit Dritten

Anmerkungen: Die letzten beiden Maßnahmen sind beispielsweise wichtig, wenn man einen *VPN*-Zugang zum Firmennetzwerk über das Internet realisieren will. Hier erfolgt erst die Anmeldung bei einem *Internet Service Provider* und dann die Verbindung zum Unternehmen. Zweites Anwendungsbeispiel wäre die Verbindung zu einem Netzwerk bei einem Kunden.

Maßnahmenkatalog *M 6 Notfallvorsorge*

- *M 6.31* Verhaltensregeln nach Verlust der Systemintegrität
- *M 6.32* Regelmäßig Datensicherung
- *M 6.40* Regelmäßige Batterieprüfung-/wechsel
- *M 6.71* Datensicherung bei mobiler Nutzung des IT-Systems

Eine ausführliche Erläuterung der einzelnen Maßnahmen können wieder in [8] nachgelesen werden. Der hier vorgestellte Gefahren- und Maßnahmenkatalog ist nur eine beispielhafte Anwendung des *Grundschutzhandbuch* des *BSI*, um die fehlende Komponente *PDA* in ein Sicherheitskonzept zu integrieren. Für einen Laptop, wie auch für ein Mobiltelefon existiert im *Grundschutzhandbuch* bereits ein komplettes Sicherheitskonzept. Somit kann mit meinem Konzept für einen PDA und dem vom *BSI* definierten Konzept für ein Mobiltelefon auch ein Konzept für ein *Smartphone* realisierbar.

Dass die mobilen Endgeräte nicht vollständig in den Anforderungen für ein Sicherheitszertifikat enthalten sind, ist heute meiner Meinung nach nicht mehr tragbar. Diese Endgeräte gewinnen immer mehr an Bedeutung und kommen in fast jedem Unternehmen in zahlreicher Ausführung zum Einsatz. Hier ist eine deutliche Nachbesserung der Zertifizierungsstellen notwendig.

Kapitel 7

Handlungsempfehlungen

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

7.1 Einführung einer einheitlichen Passwort-Policy

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

7.2 Einheitlichkeit bei mobilen Endgeräten

Die Gefahren, die von mobilen Endgeräten ausgehen, zu minimieren, ist eine einheitliche zentrale Administrierung von Synchronisationsverfahren und Sicherheitseinstellungen unumgänglich. Die in Kapitel 5 ab Seite 49 getesteten Lösungungen bieten überzeugende Möglichkeiten eine solche zentrale Administrierung durchzuführen.

An dieser Stelle möchte ich meine Empfehlung von *Pylon Anywhere* von *Sybase* als Lösung für „Datensynchronisation, Backup und Datenverteilung“, sowie den *Trusted Mobility Server* von *Trust Digital* als Sicherheitslösung noch einmal wiederholen. Dennoch haben beide Produkte in ihrer Leistungsfähigkeit überzeugt und bieten ein attraktives Lizenzmodell.

Derzeit werden Mobiltelefone bzw. Smartphones und PDAs von unterschiedlichen Abteilungen zur Verfügung gestellt. So verwaltet bei der *EDAG Engineering & Design AG* die *Bau-Abteilung* alles, was mit Telefonie (Festnetz, Mobiltelefonie) zu tun hat und die *IT-Abteilung* die PDAs. Die Zuständigkeit für Smartphones kann dadurch nicht eindeutig definiert werden, da es sich hier sowohl um ein Mobiltelefon, als auch um einen PDA handelt. Es ist somit eine einheitliche Stelle einzurichten, die sowohl Mobiltelefone, als auch PDAs verwaltet, was auch die Zuständigkeit für *Smartphones* klärt.

Durch die vorgestellten Lösungen für mobile Endgeräte können zwar viele unterschiedliche Endgeräte eingesetzt werden, dennoch sollte eine Standardisierung auf ein einheitliches Endgerät statt finden. Da die oben genannten Lösungen mit unterschiedlichen mobilen Betriebssystemen zusammen arbeiten, kann für jeden Typ (Mobiltelefon, PDA und Smartphone) ein mobiles Endgerät als Standardgerät definiert werden, aus denen der Benutzer eines auswählen kann. Diese Standardisierung auch nur auf einen Endgerätetyp beschränkt werden, um die Auswahl möglichst klein zu halten. Als Endgerätetyp ist hier sicherlich das *Smartphone* die richtige Wahl. Es bietet alle Funktionen eines Mobiltelefons und eines PDAs und vermeidet somit eine umständliche Handhabung mit zwei unterschiedlichen Endgeräten. Smartphones gibt es für unterschiedliche mobile Betriebssysteme. Für jedes dieser Betriebssysteme sollte ein einheitliches *Smartphone* definiert werden, was den Aufwand für Administrierung und Wartung erheblich minimiert. Bei Laptops und Desktop-Systemen findet eine solche Standardisierung schon seit längerem statt. Als mobiles Endgerät ist lediglich der *Palm OS*-basierte *Sony Clié* als Standardgerät definiert. Für ein Standard-*Smartphone* bieten sich für *Palm OS* der *Handspring Treo*, für *Windows CE* die *MDA*-Reihe von *T-Mobile* und für *Symbian OS* der *Nokia Communicator* oder *Sony Ericsson P800/P900* an. Eine Empfehlung für ein spezielles mobiles Betriebssystem kann ich an dieser Stelle nicht aussprechen, weil es beispielsweise von Seiten der Sicherheit dieser Systeme kein wirklich zufriedenstellendes Betriebssystem existiert[1].

Alle Smartphones gibt es in unterschiedlichen Ausstattungsvarianten. Bei der Auswahl des geeigneten Endgerätes sollte darauf geachtet werden, dass das bereits bestehende Sicherheitskonzept beachtet wird. So sollte das Smartphone keine eingebaute Digitalkamera haben und auch die Unterstützung von *WLAN* und/oder *Bluetooth* muss nicht zwangsweise vorhanden sein. Für ein so definiertes Gerät muss dann eine Gefahrenseinschätzung mit resultierenden Schutzmaßnahmen vorgenommen werden, wie ich es in Kapitel 6.3 ab Seite 80 beispielhaft aufgezeigt habe. Ist dadurch sicher gestellt, dass der Einsatz des mobilen Endgerätes ausreichend geschützt ist, so steht der Integration in ein bestehendes Sicherheitskonzept nichts mehr im Weg. Bei der nächsten Sicherheitsüberprüfung im Rahmen der Zertifizierung nach *BS 7799* kann mit diesem erweiterten Sicherheitskonzept auch der sichere Einsatz von mobilen Endgeräten gewährleistet werden.

[1] *siehe* Kapitel 2.2 ab Seite 6

(a) MDA III

(b) Handspring Treo

(c) Sony Cliè

(d) Nokia 9500 Communicator

(e) Sony Ericsson P910

Abb. 7.1: *Beispiele für geeignete Smartphones*

7.3 Erweiterter Schutz der mobilen Endgeräte

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

7.4 Sicherheit bei der Kommunikation

Auch bei der *EDAG Engineering & Design AG* ist *eMail* ein wichtiges Kommunikationsmittel. Darüber werden Termine mit Kunden vereinbart und Informationen zu aktuellen Projekten ausgetauscht. Oftmals handeln es sich hierbei auch um vertrauliche Informationen, die nicht für jeden gedacht sind. *eMail* als Kommunikationsmittel ist jedoch als sehr unsicher einzustufen, da alle Daten in der Regel im Klartext übertragen werden. Nur vereinzelt werden bei der *EDAG Engineering & Design AG* Verschlüsselungssysteme nach *PGP* eingesetzt, eine einheitliche Regelung für die gesicherte eMail-Kommunikation existiert nicht. Zwar bietet die eingesetzte Groupware *Lotus Notes* die Möglichkeit der eMail-Verschlüsselung, aber dadurch ist nicht unbedingt gewährleistet, dass der Empfänger die eMail dann auch wieder entschlüsseln kann.

Die *EDAG Engineering & Design AG* sieht ihre Stärke darin, sich auf die Bedürfnisse der Kunden einzustellen. Deswegen sollte eine Lösung zur eMail-Verschlüsselung eingesetzt werden, die alle gängigen Verschlüsselungsverfahren unterstützt. In Betracht kommen hier einmal *PGP* und die Verschlüsselung nach *S/MIME*. Bei beiden Verfahren findet eine Verschlüsselung mit öffentlichen und geheimen Schlüsseln statt und setzen eine *PKI* voraus. Als mögliche Lösung käme hier *CryptoEx* der Offenbacher Firma *Glück & Kanja* in Frage. Der *CryptoEx*-Server wird dabei als eMail-Gateway in der *DMZ* eingebunden und übernimmt vollautomatisch die Ver- und Entschlüsselung der eMails. Dazu sind Regeln definierbar, welche eMail zu welchem Empfänger in welcher Weise verschlüsselt wird. *CryptoEx* unterstützt hierbei, wie gefordert *PGP* und *S/MIME*. Fordert ein Kunde A beispielsweise eine Verschlüsselung nach *PGP* und ein Kunde B die Verschlüsselung nach *S/MIME*, so wird das im *CryptoEx*-Server durch eine Regel definiert. Schickt nun ein Mitarbeiter eine eMail an den Kunden A und an den Kunden B, so übernimmt der *CryptoEx*-Server automatisch die richtige Verschlüsselung der eMail, ohne dass der Mitarbeiter etwas davon mitbekommt. Der Mitarbeiter kann somit nicht aus Versehen eine eMail mit vertraulichen Informationen unverschlüsselt versenden, weil diese Entscheidung einzig und alleine der eMail-Gateway übernimmt.

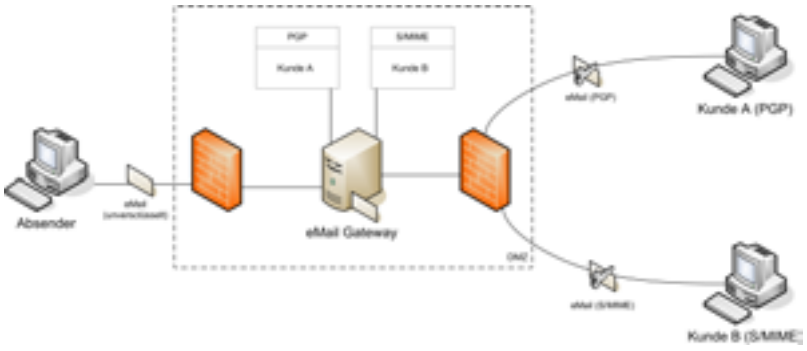


Abb. 7.2: *Beispielhafter Aufbau einer Gateway-eMail-Verschlüsselung*

Zusätzlich kann auch schon in die Standardsignatur von *Lotus Notes* ein Hinweistext am Ende der eMail eingebunden werden, der den Kommunikationspartner darauf hinweist, dass

in der eMail vertrauliche Informationen enthalten sein können, die nicht an Dritte weitergegeben werden dürfen. Zusätzlich wird der eMail-Empfänger darauf hingewiesen vertrauensvoll mit diesen Informationen umzugehen. Eine vollständige Rechtsverbindlichkeit dieses Hinweistextes besteht allerdings nicht bzw. ist letztendlich noch über eine Rechtsstelle zu prüfen. Der Hinweis zeigt aber beispielsweise gegenüber einen Kunden, dass man sich bei der *EDAG Engineering & Design AG* mit dem Thema *eMail-Sicherheit* ernsthaft auseinander setzt. Bei Finanzdienstleistern, also z.B. Banken, ist ein solcher Hinweistext schon seit längerem standardmäßig an jede ausgehende eMail angehängt. Ein entsprechender Hinweistext kann dabei wie folgt aussehen:

„*Hinweis auf Vertraulichkeit:* Diese eMail kann vertrauliche Informationen enthalten. Wenn Sie nicht der richtige Adressat sind oder diese eMail irrtümlich erhalten haben, informieren Sie bitte sofort den Absender und vernichten Sie diese eMail. Das unerlaubte Kopieren sowie die unbefugte Weitergabe dieser eMail oder von Teilen dieser eMail ist nicht gestattet. Wir haben alle verkehrsüblichen Maßnahmen unternommen, um das Risiko der Verbreitung virenbefallener Software oder eMails zu minimieren, dennoch raten wir Ihnen, Ihre eigenen Virenkontrollen auf alle Anhänge an dieser Nachricht durchzuführen. Wir schließen außer für den Fall von Vorsatz oder grober Fahrlässigkeit die Haftung für jeglichen Verlust oder Schäden durch virenbefallene Software oder eMails aus. Jede von der *EDAG Engineering & Design AG*versendete eMail ist sorgfältig erstellt worden, dennoch schließen wir die rechtliche Verbindlichkeit aus; sie kann nicht zu einer irgendwie gearteten Verpflichtung zu Lasten der *EDAG Engineering & Design AG*ausgelegt werden.“

7.5 Sensibilisierung der Mitarbeiter

HINWEIS:
Dieser Teil wurde aus datenschutzrechtlichen Gründen dieser Diplomarbeit entnommen. Der Inhalt ist nicht für Dritte bestimmt und die Veröffentlichung Bedarf der gesonderten Zustimmung des Autors bzw. der *EDAG Engineering & Design AG*. Für Fragen zu den entnommenen Inhalten steht Ihnen der Autor dieser Diplomarbeit über die im Titelblatt angegebene eMail-Adresse gerne zur Verfügung.

Eine aktive Mitarbeiterinformation fängt mit einer Schulung an, bei der über alltägliche Gefahren wie z.B. Spam und *MalWare*, aber auch der Notwendigkeit von sicheren Passwörtern informiert werden muss. Dadurch erreichen die Mitarbeiter ein Gefühl davon, wie sich solche Gefahren auswirken können. Eine umfassende Schulung aller Mitarbeiter weltweit kann hierbei recht einfach über eine eLearning-Plattform im Intranet realisiert werden, die jeder Mitarbeiter absolvieren muss. Welche Informationen für eine solche eLearning-Plattform sinnvoll sind, kann man beispielsweise auf der Webseite `http://www.bsi-fuer-buerger.de` des *BSI* finden. Die dort dargestellten Informationen sind durchaus ausreichend, um ein Grundverständnis von IT-Sicherheit zu vermitteln. Evt. sollten auch Sicherheitsinformationen vermittelt werden, die speziell für die *EDAG Engineering & Design AG* gelten. So bietet sich beispielsweise ein Lernmodul für den sicheren Einsatz von mobilen Endgeräten an, das ein Mitarbeiter absolvieren muss, bevor es ein mobiles Endgerät erhält. Dadurch kann nach und nach ein vollständiges Sicherheitsportal entstehen, in dem mittels *eLearning* Grund-

wissen vermittelt und durch Newsmeldungen über aktuelle Gefahren informiert wird. Jedes eLearning-Modul sollte dabei am Ende einen Fragebogen enthalten, den der Mitarbeiter ausfüllen muss und der z.B. vom *CSO* ausgewertet wird. Dieser Fragebogen gilt als Nachweis, dass der Mitarbeiter alle nötigen Schulungen absolviert hat und somit Grundkenntnisse in IT-Sicherheit erworben hat. Dieser Nachweis wird dazu in der Personalakte des Mitarbeiters abgelegt.

Über dieses Informationsportal zum Thema *IT-Sicherheit* kann der *CSO* darüber hinaus aktuelle Informationen zur aktuellen IT-Sicherheitslage bei der *EDAG Engineering & Design AG* oder im Internet ablegen. Dabei kann über neue Schutzmaßnahmen oder der Einsatz neuer Sicherheitslösungen informiert werden. Als zusätzliche Informationsquelle kann ein monatlicher Newsletter an alle Mitarbeiter vom *CSO* versendet werden, der ebenfalls Informationen rund um das Thema *Sicherheit* bzw. *IT-Sicherheit* liefert. Das Sicherheitsportal kann zusätzlich Mitarbeitern eine Plattform bieten, um über aktuelle Sicherheitsregelungen oder Gefahren zu diskutieren oder neue Gefahrenquellen an den *CSO* zu melden. Für Team- und Abteilungsleiter bzw. Sicherheitsbeauftragte der einzelnen Standorte können zusätzlich Schulungen zum Thema *IT-Sicherheit* angeboten werden. Dadurch erreichen diese ein erweitertes Wissen zu diesem Thema und werden dadurch zu lokalen Ansprechpartnern in den einzelnen Abteilungen und Standorten. Die Koordination dieser Ansprechpartner kann wiederum vom *CSO* übernommen werden. Durch einen solchen Prozess wird ein aktiver Umgang mit dem Thema *Sicherheit* erreicht, der sich auch mit den Anforderungen des *BS 7799-2:2002* deckt.

Sicherheit wird für ein Unternehmen wie die *EDAG Engineering & Design AG*, die täglich mit vertrauenswürdigen Informationen umgeht, immer ein akutes Thema sein. Auch die Bedrohung durch Gefahren, die durch neue Technologien oder neue Angriffsarten aus dem Internet ausgeht, wird sich nicht verringern, sondern eher um ein Vielfaches steigern. Durch die große Anzahl an Mitarbeitern weltweit ist eine aktive Sensibilisierung für neue Bedrohungen auf die Sicherheit des Unternehmens unabdingbar. Diese Sensibilisierung muss dabei in regelmäßigen Abständen durchgeführt werden und sich den unterschiedlichen Sicherheitsbedürfnissen im Unternehmen anpassen. Eine Empfehlung wäre deswegen, dass die Stelle eines hauptamtlichen *CSO* eingerichtet wird, der sich ausschließlich mit den Themen *Sicherheit* und *IT-Sicherheit* auseinander setzt. Diese Themen sind heute zu umfangreich, dass sie als Nebentätigkeit behandelt werden können. Dadurch werden auch die Forderungen des *BS 7799-2:2002* im Bereich der aktiven Mitarbeitersensibilisierung und -information besser umgesetzt und steigert langfristig den sicheren Umgang mit vertraulichen Informationen.

Schlußwort

Mobile Endgeräte sind im Unternehmensumfeld nicht mehr wegzudenken, das haben meine Recherchen zu dieser Diplomarbeit deutlich gezeigt. Kaum ein Unternehmen sieht diese Endgeräte allerdings als Bedrohung und trifft kaum Schutzmaßnahmen. Dabei sind die Gefahren durch die geringe Größe der Endgeräte und die wachsenden technischen Möglichkeiten nicht zu unterschätzen. Leicht können PDAs für Angriffe auf das Firmennetzwerk und die darin gespeicherten Informationen benutzt werden. Lösungen für IT-Sicherheit mobiler Endgeräte gibt es auf dem Markt genügend, jedoch sind diese kaum einsetzbar, wenn man eine breite Masse an mobilen Systemen unterstützen möchte.

Auch in Standardisierungsprozessen, wie z.B. *BS 7799* oder *BSI-Grundschutz*, werden mobile Endgeräte bislang stark vernachlässigt. Nur Mobiltelefone und Laptops finden hier Beachtung, weil diese mit normalen Telefonen bzw. Desktop-PCs vergleichbar sind. PDAs und Smartphones werden aber weitesgehend außer Acht gelassen. Hier ist dringend Handlungsbedarf von den Stellen gefordert, die diese Standards definieren. Die Gefahreneinschätzung und dazugehörige Schutzmaßnahmen in das *Grundschutzhandbuch* des *BSI* aufzunehmen, sollten kein größeres Problem darstellen. Andernfalls sollte unbedingt von dem Unternehmen, dass sich zertifizieren lassen will, die Initiative ausgehen, dass diese Endgeräte bei der Sicherheitsüberprüfung mit beachtet werden müssen.

IT-Sicherheit, vor allem von mobilen Endgeräten, wird in naher Zukunft verstärkt ein Thema sein. Viren, Würmer und Trojaner für diese Systeme stehen erst am Anfang ihrer Entwicklung und werden sich schnell weiter entwickeln und verbreiten.

Dennoch ist eine aktive Sicherheitspolitik in der IT kaum in Unternehmen zu finden. Meist sind Regelwerke definiert, die zwar eine Zertifizierung nach einem anerkannten Sicherheitsstandard rechtfertigen, die aber nicht aktuell den momentanen Gegebenheiten und Bedrohungen angepasst werden. Vor allem in der Sensibilisierung der Mitarbeiter sind, zumindest bei der *EDAG Engineering & Design AG*, erhebliche Defizite festzustellen. Grund hierfür sind u.a. Einsparungen für das Segement *IT-Sicherheit* und das Fehlen eines Hauptverantwortlichen, der sich ausschließlich mit diesem Thema beschäftigt. Insbesondere mittelständische und große Unternehmen dürfen aber genau dieses Segment, auch über den Einsatz von Firewalls und Virenscannern hinaus, nicht vernachlässigen.

Michael Ruck, 14 49 95

Glossar und Abkürzungen

AES	**A**dvanced **E**ncryption **S**tandard
API	**A**pplication **P**rogramming **I**nterface
APT	**A**dvanced **P**ackage **T**ool Paket-Managementsystem unter *Debian GNU/Linux*
ARM	**A**dvanced **R**ISC **M**achines Ltd. britischer Computer-Hersteller
Basel II	Vereinbarung auf Verfahren u.a. bei der Kreditvergabe durch Finanzdienstleister, bei der auch die Sicherheitspolitik in der IT eine große Beachtung für die Vergabe spielt. Hat ein Unternehmen keine ausreichende Sicherheitspolitik, so kann die Zuteilung eines Kredites verwehrt werden. Weitere Infos zu *Basel II*: *siehe* `http://www.basel-ii.info`
BIOS	**B**asic **I**nput **O**utput **S**ystem
CC	**C**ommon *C*riteria for Information Technology Security Ecaluation Internationaler Standard für Sicherheit für IT-Systeme
CPM	**C**ryptographic **P**rovider **M**anager *API* in *Palm OS Cobalt* für Sicherheitsfunktionen. Weiter Infos unter [21]
CTCPEC	**C**anadian **T**rusted **C**om**p**uter **E**valuation **C**riteria
DES	**D**ata **E**ncryption **S**tandard Verschlüsselungsverfahren, welches heute als unsicher angesehen wird. Es existieren davon mehrere Verbesserungen, wie z.B. das *3DES*
DoS	**D**enial **o**f *S*ervice Angriff auf ein IT-System bei dem das System solange belastet wird bis es vollständig ausfällt.
EAP	**E**xtensible **A**uthentication **P**rotocol
EFS	**E**ncrypted **F**ile **S**ystem Teil von NTFS5. Dient zur Verschlüsselung von Daten und Datenträgern auf einem *Windows 2000/XP Professional* Computer
FAT	**F**ile **A**llocation **T**able Ein für DOS entwickeltes Dateisystem, das noch heute in Microsoft Windows Systemen zum Einsatz kommt. Gängige Versionen: FAT, FAT16, FAT32 (für Speicher mit mehr als 2GB -kapazität)

GPL	GNU **G**eneral **P**ublic **L**icense Lizenzmodell für Open-Source-Software. Jede Software, die unter dieser Lizenz veröffentlicht wird, darf frei benutzt und kopiert werden. *siehe* `http://de.wikipedia.org/wiki/GNU_General_Public_License`
GPRS	**G**eneral **P**acket **R**adio **S**ervice Erweiterung des GSM-Standards um eine schnellere paketorientierte Datenübertragung
GSM	**G**lobal **S**ystem for **M**obile Communication Übertragungstechnik im Mobilfunk
GUI	**G**raphical **U**ser **I**nterface
HAL	**H**ardware **A**bstraction **L**ayer Architektur von Betriebssystemen um den Zugriff auf vorhandene Hardware durch Applikationen einheitlich zu gestalten. Die Applikationen müssen keine Rücksicht auf die unterschiedliche Hardware von z.B. Druckern nehmen.
HSCSD	**H**igh **S**peed **C**ircuit **S**witched **D**ata Erweiterung des GSM-Standards um eine schnellere leitungsvermittelte Datenübertragung
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers Weltweiter Verband von Ingenieuren der Elektrotechnik und Informatik. Hat ca. 360.000 Mitglieder in ca. 150 Staaten.
IKE	**I**nternet **K**ey **E**xchange Schlüsselverwaltung und -austausch im *IPSec*-Protokoll
IrDA	**I**fra**r**ed **D**ata **A**ssociation
ISDN	**I**ntegrated **S**ervices **D**igital **N**etwork
ITSEC	**I**nformation **T**echnology **S**ecurity **E**valuation **C**riteria
LDAP	**L**ightweight **D**irectory **A**ccess **P**rotocol
MAC-Adresse	**M**edia **A**ccess **C**ontrol Adresse Eindeutige Netzwerkadresse einer Netzwerkschnittstelle, z.B. eine Netzwerkkarte, in der Schicht 2 des OSI-Schichtenmodells. Die Adresse besteht aus sechs Kombinationen von zweistelligen hexadezimalen Zahlen.
MalWare	*MalWare* setzt sich zusammen aus *malicious* und *Software* und bedeutet so viel wie *fehlerhafte Software*
Man-in-the-Middle	Gängige Hacker-Praxis um eine Kommunikationsverbindung abzuhören und evt. zu beeinflussen. Dabei schaltet sich der Hacker direkt zwischen die beiden Opfer.
MD5	**M**essage **D**igest **5** Weit verbreitetes Verfahren zum Bilden von Hash-Werten
MIDP	**M**obile **I**nformation **D**evice **P**rofile Profil der *Java 2 Micro Edition* Plattform, das speziell auf die Fähigkeiten mobiler Endgeräte ausgelegt ist.

MIPS	**M**icroprocessor without **i**nterlocked **p**ipeline **s**tages Mikroprozessor mit RISC-Prozessorarhcitektur ohne Pipeline-Sperren; Entwickelt 1981 von John Hennessy an der Stanford University. Einsatz in SGI UNIX-Workstations und mobilen Endgeräten
MS-DOS	**M**icro**s**oft - **D**isk **O**perating **S**ystem Erstes, noch textbasiertes, Betriebssystem von Microsoft auf 16bit Basis, entwickelt 1981.
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NSA	**N**ational **S**ecurity **A**gency
NT	**N**ew **T**echnology Betriebssystemarchitektur von Microsoft
NTFS5	**N**ew **T**echnology **F**ile **S**ystem Dateisystem für NT-basierte Windows-Betriebssysteme. NTFS5 kommt seit Windows 2000 zum Einsatz und vorallem für den Einsatz großer Partitionen empfehlenswert.
NTLM	**N**ew **T**echnology **L**AN-**M**anager Authentifizierungs-Verfahren bei Windows NT-basierten Betriebssystemen
OBEX	**Ob**ject **Ex**change
OPIE	**O**pen **P**almtop **I**ntegrated **E**nvironment Auf *Debian GNU/Linux*-basiertes Betriebssystem für PDAs und Smartphones
PCMCIA	**P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation Standardisierte Schnittstelle für Erweiterungen bei mobilen Endgeräten
PDA	**P**ersonal **D**igital **A**ssistant *oder* **p**ersönlicher **d**igitaler **A**ssistent Kleiner tragbarer Computer, hauptsächlich zur Speicherung von PIM-Daten gedacht
PIM	**P**ersonal **I**nformation **M**anagement
PIN	**P**ersonal **I**dentification **N**umber Art eines Passworts, meist aus vier Zahlen, das zur Authentifizierung an Endgeräten, wie z.B. Mobiltelefonen, verwendet wird.
POSIX	**P**ortable **O**perating **S**ystem **I**nterface for Uni**x** Ein von der IEEE entwickelter Standard einer Schnittstelle zwischen Applikationen und Betriebssystem UNIX. (IEEE 1003, DIN/ISO 9945)
PPP	**P**oint-to-**P**oint **P**rotocol
PPPoE	**P**oint-to-**P**oint **P**rotocol **o**ver **E**thernet
PPTP	**P**oint-to-**P**oint **T**unneling **P**rotocol
RAM	**R**andom **A**ccess **M**emory
RC4	**R**on's **C**ipher **4** Kryptografisches Verfahren zur Datenverschlüsselung entwickelt von Ronald L. Rivest für RSA Security

RFC	**R**equest **f**or **C**omments Dokumente über Technik und Organisation im Internet. Eigentlich dazu gedacht Kommentare zu einzelnen Vorschlägen abzugeben, mittlerweile sind viele *RFC*-Dokumente wie Standards zu verstehen. Beispiele hierfür sind *UDP* (RFC 768), *TCP* (RFC 793), *IP* (RFC 791), *DHCP* (RFC 2131), *FTP* (RFC 959) oder *HTTP* (RFC 2616).
ROM	**R**ead-**O**nly **M**emory
RPM	**RPM** **P**ackage **M**anager
RSA	Steht für die Anfangsbuchstaben der Nachnamen der Entwickler Ronald L. **R**ivest, Adi **S**hamir und Lenoard **A**dleman
SDIO	**S**ecure **D**igital **I**nput/**O**utput Definition einer Schnittstelle für mobile Endgeräte, die neben den SD-Speicherkarten auch mit Erweiterungkarten im SD-Format umgehen kann.
SDK	**S**oftware **D**eveloper **K**it Sammlung von Programmen und Dokumentationen zu einem Produkt, mit dem Software-Entwickler neue Anwendungen entwickeln können
SHA	**S**ecure **H**ashing **A**lgorithm
SHx	**S**uper**H** RISC engine Prozessor von Hitachi/Renesas für mobile Endgeräte. Aktuelle Versionen: SH3 und SH4
SSID	**S**ervice **S**et **Id**entifier Identifikationsname für ein *Wireless LAN*
SSL	**S**ecure **S**ocket **L**ayer Von der Firma Netscape entwickeltes Übertragungsprotokoll für eine sichere Kommunikation
TKIP	**T**emporal **K**ey **I**ntegrity **P**rotocol
TLS	**T**ransport **L**ayer **S**ecurity Durch die *IETF* standardisierter Nachfolger von SSL v3
UMTS	**U**niversal **M**obile **T**elecommunications **S**ystem Neuer Mobilfunkstandard mit erweiterten multimedialen Diensten. Wird oft auch als „3G“ bezeichnet, was für die dritte Mobilfunkgeneration steht.
USB	**U**niversal **S**erial **B**us Bussystem von einem Computer mit einem externen Endgerät zum Austausch von Daten
VPN	**V**irtual **P**rivate **N**etwork
WEP	**W**ired **E**quivalent **P**rivacy Standard für die Sicherheit in einem *Wireless LAN*
Wordlist-Attacke	Attacke bei der mit einer Liste aus unterschiedlichen Zeichenketten versucht wird, ein Passwort o.ä. zu entschlüsseln.

WPA	**W**i-Fi **P**rotected **A**ccess Neuer Sicherheitsstandard für Wireless LANs bei dem *TKIP* und *EAP* zum Einsatz kommen. Ersetzt den alten Standard WEP
X.509	X.509 ist ein Standard der ITU-T für eine Public Key Infrastructure und ist in der RFC 3280 definiert. Aktuelle Version: X.509v3
x86	Prozessorbefehlssatz von Intel, der erstmals 1971 veröffentlicht wurde; Kommt in allen aktuellen PC-Prozessoren zum Einsatz; Bekannteste Vertreter: 8086, 80286 (kurz: 286), 80486 (kurz: 486), Pentium (eigentlich 80586 und 80686), Celeron, Centrino usw.
YaST	**Y**et **a**nother **S**etup **T**ool Installations- und Konfigurationswerkzeug unter *SuSE Linux*. Seit der Übernahme von *SuSE* durch *Novell* unter der *GPL* gestellt.

Literaturverzeichnis

[1] Blunk, L. ; Vollbrecht, J. : PPP Extensible Authentication Protocol (EAP), RFC 2284 / Merit Network. 1998. ↪ *siehe* `http://www.faqs.org/rfcs/rfc2284.html`

[2] Borchers, D. : Apples Newton: Der Schwerkraft getrotzt, doch der Zeit voraus. In: *Heise Newsticker* (2003), August. ↪ *siehe* `http://www.heise.de/ct/aktuell/meldung/39105`

[3] British Standard: *BS 7799-2:2002 : Information security management systems - Specifications with guidance for use.* British Standard, September 2002

[4] Bundesamt für Sicherheit in der Informationstechnik: *Common Criteria (ISO/IEC 15408) - Teil 1: Einführung und allgemeines Modell.* Version 2.1. Bonn: Bundesamt für Sicherheit in der Informationstechnik, August 1999

[5] Bundesamt für Sicherheit in der Informationstechnik: *Common Criteria (ISO/IEC 15408) - Teil 2: Funktionale Sicherheitsanforderungen.* Version 2.1. Bonn: Bundesamt für Sicherheit in der Informationstechnik, August 1999

[6] Bundesamt für Sicherheit in der Informationstechnik: *Common Criteria (ISO/IEC 15408) - Teil 3: Anforderungen an die Vertrauenswürdigkeit.* Version 2.1. Bonn: Bundesamt für Sicherheit in der Informationstechnik, August 1999

[7] Bundesamt für Sicherheit in der Informationstechnik: Bluetooth - Gefährdungen und Sicherheitsmaßnahmen / Bundesamt für Sicherheit in der Informationstechnik. 2003. ↪ *siehe* `http://www.bsi.de/literat/doc/bluetooth/`

[8] Bundesamt für Sicherheit in der Informationstechnik: *IT-Grundschutzhandbuch: 5. EL Stand Oktober 2003.* Auflage 5. Bundesamt für Sicherheit in der Informationstechnik, Oktober 2003. ↪ *siehe* `http://www.bsi.bund.de/gshb/deutsch/download/GSHB2003.pdf`

[9] Dierks, T. ; Karlton, P. : The TLS Protocol, RFC 2246 / Certicom, Netscape Communications. 1999. ↪ *siehe* `http://www.faqs.org/rfcs/rfc2246.html`

[10] Dixon, K. : Symbian OS Version 7.0s functional description / Symbian Ltd. 2003. ↪ *siehe* `http://www.symbian.com/technology/symbos-v7s-det.html`

[11] Eastlake, D. ; Jones, P. : U.S. Secure Hash Algorithm 1 (SHA1), RFC 3174 / Request for Comments Network Working Group. 2001. ↪ *siehe* `http://www.faqs.org/rfcs/rfc3174.html`

[12] Gartner Group: Windows CE Surpasses Palm OS in 3Q04. In: *Gartner Research Center, Mobile & Wireless* (2004), November. ↪ *siehe* http://www3.gartner.com/DisplayDocument?doc_cd=124782

[13] Hamzeh, K. ; Pall, G. S.: Point-to-Point Tunneling Protocol (PPTP), RFC 2637 / Microsoft Corporation. 1999. ↪ *siehe* http://www.faqs.org/rfcs/rfc2637.html

[14] Harkins, D. ; Carrel, D. : The Internet Key Exchange (IKE), RFC 2409 / Cisco Systems. 1998. ↪ *siehe* http://www.faqs.org/rfcs/rfc2409.html

[15] Kent, S. ; Atkinson, R. : Security Architecture for the Internet Protocol / BBN Corporation. 1998. ↪ *siehe* http://www.faqs.org/rfcs/rfc2401.html

[16] Melnik, D. ; Dinman, M. ; Muratov, A. : *PDA Security - Incorporating Handhelds into the Enterprise*. Erste Auflage. The McGraw Hill Corporation, USA, 2003. ↪ *siehe* http://www.pdasecurity-book.com. – ISBN 0–07–142490–3

[17] Microsoft Corporation: Microsoft Windows Embedded Developer Center. In: *Microsoft MSDN* (2004). ↪ *siehe* http://msdn.microsoft.com/embedded/

[18] NIST: Secure Hash Standard, FIPS PUB 180-1 / National Institute of Standards and Technology. 1995. ↪ *siehe* http://www.itl.nist.gov/fipspubs/fip180-1.htm

[19] NIST: Advanced Encryption Standard (AES), FIPS PUB 197 / National Institute of Standards and Technology. 2001. ↪ *siehe* http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[20] NIST: Secure Hash Standard, FIPS PUB 180-2 / National Institute of Standards and Technology. 2002. ↪ *siehe* http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchange notice.pdf

[21] PalmSource: PalmOS Cobalt Security / Palm Source Inc. 2003. ↪ *siehe* http://www.palmos.com/dev/support/docs/protein_books/SecurityAnd Cryptography/SecurityConcepts.html

[22] Rescola, E. : Diffie-Hellman Key Agreement Method, RFC 2631 / RTFM Inc. 1999. ↪ *siehe* http://www.faqs.org/rfcs/rfc2631.html

[23] Rijmen, V. : Efficient Implementation of the Rijndael S-box. In: *Katholieke Universiteit Leuven* (2000). ↪ *siehe* http://www.esat.kuleuven.ac.be/ rijmen/rijndael/sbox.pdf

[24] Rivest, R. L.: The MD5 Message-Digest Algorithm, RFC 1321 / MIT Laboratory for Computer Science. 1992. ↪ *siehe* http://www.faqs.org/rfcs/rfc1321.html

[25] Ruck, M. : Vodafone Mobile Connect Card UMTS Review / EDAG Engineering & Design AG. 2004

[26] Russinovich, M. : Inside Encrypting File System, Part 1. In: *Windows IT Pro* (1999), Juni. ↪ *siehe* http://www.win2000mag.com/Article/ArticleID/5387/5387.html

[27] Russinovich, M. : Inside Encrypting File System, Part 2. In: *Windows IT Pro* (1999), Juli. ↪ *siehe* http://www.win2000mag.com/Article/ArticleID/5592/5592.html

[28] Schneier, B. : Analysis of Microsoft PPTP Version 2 / Counterpane Systems. 1998. ↪ *siehe* http://www.schneier.com/pptp.html

[29] Wang, X. ; Feng, D. ; Lai, X. ; Yu, H. : Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD / Dept. of Computer Sciences and Engineering, Shanghai Jiaotong University, Shanghai. 2004. ↪ *siehe* `http://eprint.iacr.org/2004/199.pdf`

[30] Zimmermann, K. : *Der Rijndael (AES) Algorithmus*, Universität Ulm, Fachbereich Technische Informatik, Diplomarbeit, Juli 2004

Abbildungsverzeichnis

Index

Anhang A

Test der *Vodafone Mobile Connect Card UMTS*

Michael Ruck, 14 49 95

A.1 Einleitung

Vodafone bietet schon seit längerem eine *Mobile Connect Card* an, die eine Datenkommunikation über *GPRS* ermöglicht. Diese Technik ist es somit für Unternehmen interessant, dessen Mitarbeiter viel unterwegs sind, wie z.B. Außendienstmitarbeiter oder freie Mitarbeiter. Diese können sich über das *GSM* Mobilfunknetz, welches weltweit als genormter Standard vorhanden ist, mit dem Firmennetzwerk verbinden. *GPRS* ist neben *HSCSD* eine Erweiterung des *GSM*-Standards, die die Geschwindigkeit der Datenübertragung erhöht. So hat man bei einer normalen GSM-Verbindung eine nutzbare Datenrate von 9,6 kbit/s. Bei der GPRS-Erweiterung kommt man theoretisch auf eine Datenrate von 171,2 kbit/s, wobei in der Praxis im Mittel nur 53,6 kbit/s übrig bleiben.

Um die Datenrate noch weiter zu erhöhen wurde vor einigen Jahren eine neue Mobilfunk-Technologie standardisiert und die Funkfrequenzen in Deutschland medienwirksam verteilt, das *UMTS*. Dieses neue System sendet in einem höheren Frequenzband (1.900 bis 2.170 MHz)[1] und liefert eine Datenrate von 384 kbit/s[2], theoretisch sind bis zu 2 MBit/s möglich. In wiefern sich diese Bandbreite in der Praxis zu erreichen, ist bleibt abzuwarten, da momentan nur wenige Teilnehmer im UMTS-Netz registriert sind und der Aufbau des flächendeckenden Empfangs noch im Gange ist[3]. Dass dies ein langwieriger und aufwendiger Prozess liegt unter anderem auch daran, dass die einzelnen UMTS-Zellen[4] kleiner sind als die GSM-Zellen[5] und somit mehr Technik installiert werden muss.

In vielen großen Regionen ist *UMTS* schon verfügbar und somit beginnen die Mobilfunkanbieter *UMTS* als neue Technik anzupreisen. *Vodafone* bietet speziell für den Business-Bereich eine neue Version ihrer *Mobile Connect Card*, die sowohl die *GSM*-Erweiterung *GPRS*, als auch die neue Technologie *UMTS* unterstützt, die *Mobile Connect Card UMTS*. Hersteller der *Mobile Connect Card UMTS* ist die belgische *OPTION Wireless Technology*, die auch für die *Vodafone*-Konkurrenten *T-Mobile* und *orange* ähnliche Produkte herstellt.

Inwiefern der Einsatz dieser *Mobile Connect Card UMTS* insbesondere in einer kleinstädtischen Region wie Stadt und Landkreis Fulda heute schon sinnvoll ist soll dieser Test zeigen.

A.2 Lieferumfang und Testumgebung

Die Verpackung der *Mobile Connect Card UMTS* besteht aus einer stabilen Box in der Größe einer Doppel-DVD-Hülle. Darin befindet sich das ca. 20 Seiten starke Handbuch

[1] *GSM* in Europa: bei 900 MHz bzw. 1.800 MHz, *GSM* in Amerika: 900 MHz bzw. 1.900 MHz
[2] nur Downlink, Uplink liegt bei 64 kbit/s
[3] Informationen zu bereits versorgten Gebieten liefert Vodafone unter
`http://www.vodafone.de/business/support/45255.html`
[4] Eine Zelle ist der Bereich in dem ein Mobilfunkmast abstrahlt
[5] UMTS-Zelle: ca. 1 bis 2 km; GSM-Zelle: ca. 5 km

inkl. CD-ROM und natürlich die *Mobile Connect Card UMTS* selbst. Diese ist noch einmal separat in einer stabilen Plastik-Box eingepackt ist. Die *Mobile Connect Card UMTS* selbst ist in der Form einer *PCMCIA*-Karte und benötigt ein Gerät mit einer *PCMCIA Typ II* Schnittstelle. Die *Mobile Connect Card UMTS* unterstützt dabei die drei Frequenzbereiche 900 MHz, 1.800 MHz und 1.900 MHz und ist somit auch für den Einsatz in Amerika geeignet. Was nicht im Lieferumfang enthalten ist, ist die SIM[6]-Karte, die bei jedem Mobilfunkgeräte, egal ob *GSM* oder *UMTS* notwendig ist und weiterhin die gleiche Bauform hat. Man kann hierbei entweder eine schon vorhandene SIM-Karte nutzen oder einen neuen Mobilfunkvertrag z.B. mit *Vodafone* abschließen. Bei meinem Testgerät war eine SIM-Karte bereits enthalten. Die *Mobile Connect Card UMTS* hat zusätzlich die Möglichkeit eine externe Antenne sowohl für den *GPRS*-, als auch für den *UMTS*-Empfang anzuschließen. Somit kann die Empfangsleistung der Karte nochmal erhöht werden und ist bestimmt für Umgebungen interessant in denen der Funkverkehr beispielsweise durch Stahlträger o.ä. gestört wird.

Abb. A.1: *grün: GPRS; blau: UMTS*

Als Mindestanforderungen benötigt die *Mobile Connect Card UMTS* 32 MB RAM und ca. 50 MB an Festplattenplatz. Als Betriebssystem werden alle aktuellen Versionen von *Microsoft Windows* unterstützt. Als Testgerät wurde ein *Toshiba Satellite Pro 4600* Notebook mit einem *Pentium III* 700 MHz Prozessor, 256 MB Hauptspeicher und 10 GB Festplatte benutzt. Als Betriebssystem kam *Microsoft Windows XP Professional* zum Einsatz. Alle Beschreibungen beziehen sich direkt auf diese Testumgebung, vor allem bei Beschreibungen zum Betriebssystem.

A.3 Installation und Erster Eindruck

Die mitgelieferte CD-ROM enthält alle Treiber und Anwendungen, die für den Betrieb der *Mobile Connect Card UMTS* notwendig sind. Es ist allerdings genau darauf zu achten den

[6]Subscriber Identity Module

Anweisungen im Handbuch zu folgen. Laut dem Handbuch soll die *Mobile Connect Card UMTS* nämlich erst in den PCMCIA-Slot gesteckt werden, wenn man von der Installationssoftware dazu aufgefordert wird.

Um nun die Installation zu starten genügt es einfach die mitgelieferte CD-ROM in das CD-Laufwerk zu legen und das Installationsprogramm beginnt seine Arbeit. Das Installationsprogramm bietet nun zwei Möglichkeiten die sog. *Vodafone Mobile Connect*-Software zu installieren, die Version *Internet* und die Version *Zugang zum Firmennetz*. Der Unterschied dieser beiden Versionen liegt eigentlich nur darin, dass jeweils ein anderes Verbindungsprofil in der Software abgelegt wird. Da für die Version *Zugang zum Firmenetz* ein gesonderter Verbindungstarif mit *Vodafone* abgeschlossen werden muss, habe ich mich für die Version *Internet* entschieden und die Installation gestartet.

Es folgen ein paar Anweisungen am Bildschirm, die zu befolgen sind und danach erfolgt ein Neustart des Betriebssystems. Nach dem Neustart werden die restlichen Programmteile und Treiber installiert und man wird aufgefordert, die *Mobile Connect Card UMTS* in den *PCMCIA*-Slot zu stecken. Dabei aber nicht vergessen vorher noch die SIM-Karte auf der Rückseite in die *Mobile Connect Card UMTS* zu stecken. Wenn die *Mobile Connect Card UMTS* in dem PCMCIA-Slot steckt, fangen sofort die beiden Leuchtdioden[7] an zu blinken und signalisieren die Bereitschaft der Karte. Auch das Betriebssystem erkennt sofort die neue Karte und fängt an die Treiber zu installieren.

Abb. A.2: *Mobile Connect Card UMTS im PCMCIA-Slot eines Laptops*

Beim Start des Betriebssystems wurde die *Vodafone Mobile Connect*-Software automatisch gestartet. Diese erkennt sofort, dass die Karte eingesteckt wurde und fängt an, eine Verbindung aufzubauen. Laut Handbuch ist während diesem Verbindungsaufbau noch die Eingabe der *PIN* notwendig, wie man es bei einem Mobiltelefon gewohnt ist. In meinem Test war keine *PIN* auf der SIM-Karte gespeichert und somit war die *Mobile Connect Card UMTS* automatisch mit dem Funknetz verbunden.

Somit war die *Mobile Connect Card UMTS* innerhalb von ca. fünf Minuten auf sehr einfache Art und Weise installiert und einsatzbereit. Das Handbuch gab dabei genügend Hilfestellung und machte die Installation der *Mobile Connect Card UMTS* inkl. Treiber und Software zu einem Kinderspiel.

Anmerkung: Leider kam es bei der mitgelieferten Programmversion (1.3.10) oftmals zu Verbindungsschwierigkeiten bzw. zum kompletten Absturz des Laptops. Aus diesem Grund habe ich die aktuelle Softwareversion (3.0.2) von der *Vodafone*-Webseite heruntergeladen und installiert. Mit der neuen Version hatte ich oftmals eine bessere Verbindung zum UMTS-Netz und die Software lief wesentlich stabiler. Die neue Softwareversion unterstützt darüber hin-

[7] grün: GPRS; blau: UMTS

aus verschiedene Karten von verschiedenen Herstellern. Zusätzlich kann man mit der neuen Version auch auf WLAN-Hotspots zugreifen, insofern das von der Datenkarte unterstützt wird. Alle weiteren Funktionen bleiben allerdings gleich. In meinen Tests habe ich mich ausschließlich mit der neuen Version der *Vodafone Mobile Connect*-Software beschäftigt. Eine Installationsanweisung finden Sie im Anhang.

Abb. A.3: *Screenshot der mitgelieferten Software-Version 1.3.10*

A.4 Die *Vodafone Mobile Connect*-Software

Mit der *Mobile Connect Card UMTS* kommt eine ganze Software-Suite, die einem bei der mobilen Kommunikation unterstützt, die *Vodafone Mobile Connect*-Software. Die zentrale Oberfläche wird von Vodafone *Dashboard* genannt. In der unteren Statusleiste des *Dashboards* findet man alle wichtigen Informationen zur momentanen Verbindung. So wird angezeigt in welchem Mobilfunknetz man sich momentan befindet (z.B. *Vodafone*) und welche Technologie genutzt wird (*GPRS* oder *UMTS*). Außerdem findet man alle wichtigen Informationen zur momentanen Verbindung, wie z.B. die Dauer der aktuellen Verbindung bzw. verbrauchtes Volumen, die noch einmal etwas detaillierter aufgeschlüsselt werden. Wenn man nicht mit dem Mobilfunknetz verbunden ist, dann steht dort nur *Nicht verbunden*.

An der oberen Leiste bietet das *Dashboard* eine Zugriffsmöglichkeit auf alle unterstützten Kommunikationsmittel, wie z.B. *SMS* oder *eMail*. Interessant ist noch der Bereich *VPN*, der

Abb. A.4: *Screenshot der Software-Version 3.0.2*

standardmäßig deaktiviert ist. Er bietet die Möglichkeit, direkt aus dem *Dashboard* heraus eine gesicherte Verbindung, z.B. zum Firmennetzwerk, herzustellen. Außerdem bekommt man hier Informationen zur Nutzung von *GPRS* und *UMTS* und Zugriff auf einen Support. Über den Button *Web* wird schließlich der Standard-Webbrowser gestartet.

Die *Vodafone Mobile Connect*-Software startet standardmäßig mit der SMS-Oberfläche. Dies geschieht einfach aus dem Grund, dass dafür keine kostenpflichtige Verbindung aufgebaut sein muss, wie bei den andere Kommunikationsmitteln. Natürlich kann bei der *Vodafone Mobile Connect*-Software eingestellt werden, dass automatisch eine Verbindung zum jeweiligen Mobilfunknetz aufgebaut werden soll. Dies würde ich aber nicht aktivieren, weil sonst schnell eine ungewollte Verbindung aufgebaut wird und unnötige Kosten entstehen. Wenn beispielsweise *eMail* oder *Web* ausgewählt wird, baut sich nach eine Benutzeraufforderung eine Verbindung mit dem Mobilfunknetz auf. In der SMS-Oberfläche ist es außerdem möglich die Kontakte auf der SIM-Karte zu pflegen oder Kontakte aus *CSV*[8]-Dateien einzulesen. Somit kann man recht einfach die Telefondaten zwischen der SIM-Karte und seiner Adressverwaltung abzugleichen, vorallem sinnvoll, wenn man die SIM-Karte auch in einem Mobiltelefon einsetzt.

Kernstück für den ganzen Verbindungsaufbau ist der Profilmanager, den man über *Extras* → *Profile* erreicht. Man hat hier die Möglichkeit ein bereits vorhandenes Profil auszuwählen, es zu bearbeiten oder ein Neues zu erstellen. Während der Installation wurde in der Regel schon ein Profil erstellt, das den Namen *Vodafone Germany* hat. Dieses Profil bietet eigentlich schon alles, was man für einen erfolgreichen Verbindungsaufbau benötigt. Wenn man doch

[8]Text-Datei in der die Daten mit Semikolon oder Komma getrennt werden; Wird z.B. von Microsoft Outlook verwendet

Änderungen an dem Profil vornehmen möchte, muss man den Profilmanager starten und man wird über Dialoge durch die einzelnen Einstellmöglichkeiten geführt. Interessant für den Test ist eigentlich nur die Einstellung, welchen Verbindungstyp die Karte bevorzugen soll, also *GPRS* oder *UMTS*. Standardmäßig ist dort *UMTS bevorzugt* eingestellt, so dass die *Mobile Connect Card UMTS* immer *UMTS* auswählt, insofern ein solches Netz verfügbar ist. Andernfalls schaltet die Karte automatisch auf *GPRS* um. Wenn man diesen automatischen Wechsel unterbinden möchte, wählt man *Nur UMTS* oder *Nur GPRS* aus und die *Mobile Connect Card UMTS* benutzt nur den gewählten Verbindungstyp.

Die *Mobile Connect Card UMTS* ist im Unternehmensbereich wirklich interessant, wenn man auch auf das Firmennetzwerk zugreifen kann. Somit ist die VPN-Funktion der Software eine genauere Betrachtung wert. Standardmäßig ist dieser Punkt deaktiviert und muss erst über den Menüpunkt *Extras* → *Optionen* → *Anwendungen* → *VPN* aktiviert werden. Die einfachste Variante ist dort *MS VPN* auszuwählen. Voraussetzung hierfür ist, dass bereits eine VPN-Verbindung auf Betriebssystem-Ebene eingerichtet ist. Wie das bei der *EDAG Engineering & Design AG* geht, ist beispielsweise in der *Bedienungsanleitung EDAG VPN* beschrieben, die man bei der Beantragung eines Remote-Zuganges erhält. In dieser Anleitung wird eine *VPN*-Verbindung *EDAG VPN* angelegt, auf die man auch über die *Vodafone Mobile Connect*-Software zugreifen kann. Hat man die *VPN*-Unterstützung in der *Vodafone Mobile Connect*-Software aktiviert, so wird das Symbol *VPN* in der Buttonleiste angezeigt. Durch die Auswahl von *MS VPN* übernimmt das Betriebssystem die Kontrolle über die VPN-Verbindung.

Will man eine VPN-Verbindung mit dem Firmennetzwerk (im Test mittels *EDAG VPN*) aufbauen, so wählt sich die Karte zunächst in das Mobilfunknetz, je nach Auswahl bzw. Verfügbarkeit in *GPRS* oder *UMTS*, ein und baut danach die VPN-Verbindung auf. Den Aufbau der VPN-Verbindung erkennt man über den Windows-Dialog, der vor dem Verbindungsaufbau noch den Benutzernamen und dass Passwort[9] abfragt. Somit ist der Laptop zunächst über die *Mobile Connect Card UMTS* mit dem Internet verbunden und erstellt danach eine gesicherte Verbindung zum Firmennetzwerk.

Alle eingerichteten Profile sind auch außerhalb der *Vodafone Mobile Connect*-Software in den Windows Netzwerkverbindungen zu erreichen. So existiert dort beispielsweise eine Verbindung mit Namen *Vodafone Germany Connection*. Somit kann eine Verbindung zum Mobilfunknetz über die *Mobile Connect Card UMTS* aufgebaut werden, ohne die *Vodafone Mobile Connect*-Software zu benutzen.

A.5 Testszenarios

Nachdem alle Einstellungen vorgenommen wurden und man sich ein wenig mit der *Vodafone Mobile Connect*-Software vertraut gemacht hat, kann der eigentliche Test beginnen. Um einen umfassenden Eindruck zu bekommen, habe ich den Test in drei Bereiche aufgeteilt. Ziel dieser Aufteilung ist es in den unterschiedlichen Gebieten die Empfangsstärke und die durchschnittlich erreichte Datenrate zu dokumentieren. Im Test wurde im Profil

[9] bekommt man gemäß der Anleitung *Bedienungsanleitung EDAG VPN*

als Verbindungstyp *Nur UMTS* ausgewählt, weil eine GPRS-Verbindung eigentlich immer flächendeckend[10] möglich ist.

Beim ersten Test wird eine Verbindung mit dem Internet aufgebaut und dort verschiedene Webseiten, u.a. *Google Deutschland* (`http://www.google.de`), *Heise Online* (`http://www.heise.de`) und die Webseite der *EDAG Engineering & Design AG* (`http://www.edag.de`), abgerufen. Um eine bessere Einschätzung der durchschnittlichen Downloadrate zu bekommen wird eine ca. 4 MB große Datei[11] heruntergeladen. Dabei wird neben der durchschnittlichen Datenrate auch die Dauer des Downloads gemessen. Nach dem Verbindungsaufbau meldet *Microsoft Windows* stets eine Datenrate von 384 kbit/s, was der momentanen technischen Umsetzung entspricht. Dieser Wert ist aber ein rein theoretischer Wert, der tatsächlich erzielte Wert ist meist deutlich geringer.

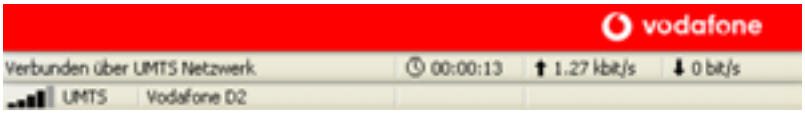


Abb. A.5: *Statusleiste bei Verbindung zum UMTS-Netzerk*

Abb. A.6: *Anzeige von Windows bei Verbindung zu UMTS*

Interessanter Weise bekommt man eine IP-Adresse aus dem Adressraum 10.0.0.0 bis 10.255.255.255 dynamisch zugewiesen, wenn man sich im UMTS-Netz einwählt. Dies entspricht einem privaten Class-A-Netz, ähnlich den IP-Adressen 192.168.*.* im Class-C-Netz, die im Smalloffice-Bereich gerne eingesetzt werden. Auch wird jede Internet-Anfrage über acht Abschnitte geleitet, die sich alle in privaten Netzen befinden, bevor überhaupt die Anfrage dem Internet übergeben wird.

Als zweiter Test wird eine VPN-Verbindung zum Netzwerk der *EDAG Engineering & Design AG* aufgebaut und dort auf das Intranet und meine *Lotus Notes*-Mailbox über den *Lotus Notes*-Cliebt zugegriffen. Außerdem wird versucht auf das Home-Verzeichnis im Netzwerk zuzugreifen. Auch hier wird wieder die durchschnittliche Datenrate aufgezeichnet.

Die beiden Tests werden in unterschiedlichen räumlichen Umgebungen durchgeführt, da es je nach Ausbau der UMTS-Verfügbarkeit zu starken Schwankungen kommen kann. Im Bereich *EDAG TEC-Center* wird der Standort des Hauptsitzes der *EDAG Engineering & Design AG* näher betrachtet. Vor allem die Erreichbarkeit des *UMTS*-Netzes und die dadurch erzielten Datenraten sind hier interessant. Im Bereich *Innenstadt Fulda*, in dem sich

[10] laut Vodafone: 96% von Deutschland sind abgedeckt
[11] URL zur Testdatei: `http://seclab.download.cwsnet.de/allgemeines/buecher/Kryptologie.zip`

die beste Ausbaustufe hier in der Region befindet, wird die Verbindung einem großen Lasttest unterzogen, um die Stabilität und Schnelligkeit der UMTS-Verbindung herauszubekommen. In der letzten Umgebung *Randgebiet Fulda* wird lediglich die Netzabdeckung in den Randgebieten der Stadt aufgezeichnet. Hier ist festzustellen, wie gut der Ausbau des *UMTS*-Netzes in der Stadt Fulda zum heutigen Zeitpunkt ist.

Anmerkung: Die Empfangsstärke des jeweiligen Netzes wird in diesem Test in *Balken* angegeben. Wobei *0 Balken* „keine Verbindung“ und *5 Balken* „voller Empfang“ bedeuten. Dies mag eine ungewöhnliche Maßangabe sein, aber leider fehlen mir die entsprechenden technischen Hilfsmittel um die Empfangsstärke genau zu messen. Die *Vodafone Mobile Connect*-Software zeigt aber in seiner Oberfläche die Empfangsstärke, ähnlich wie bei Mobiltelefonen, in kleinen Balken an. Diese Angabe dürfte jedem Mobilfunknutzer bekannt sein und wird somit als bekannte Maßeinheit angenommen.

A.5.1 EDAG TEC-Center

Eigentlich ist es nicht nötig auf dem Geländer der *EDAG Engineering & Design AG* extra eine VPN-Verbindung zum Firmennetzwerk aufzubauen. Man hat hier besser die Möglichkeit über einen normalen Anschluß mit einem Netwerkkabel an das Netzwerk zu kommen. Jedoch sollte schon bei der Installation der Karte auch die Funktionalität inkl. Verbindungsaufbau usw. getestet werden. Das geschieht in der Regel im *TEC-Center* der *EDAG Engineering & Design AG*, in dem sich die *EDV-Abteilung* befindet. Natürlich finden die Tests ohne eingestecktem Netzwerkkabel statt, um durch die Verbindung zum Internet keine zusätzliche Schwachstelle für das Netzwerk darzustellen.

Leider gibt es im *TEC-Center* schon ein kleines Problem, denn die Netzabdeckung für dieses Gebiet scheint noch nicht ausreichend zu sein. Manchmal erhält man nur ein Netz, wenn man die Karte kurz aus dem PCMCIA-Slot heraus nimmt und sie dann wieder rein steckt oder aber die *Vodafone Mobile Connect*-Software bzw. den Laptop komplett neu startet. Dann bekommt man meist eher eine Verbindung zu *GPRS* anstatt zu *UMTS*. Wenn man nur das *UMTS*-Netz akzeptiert, zeigt die *Vodafone Mobile Connect*-Software ständig *suche Netzwerk* an. Wurde jedoch eine Verbindung zum *UMTS*-Netz aufgebaut, so schwankt die Empfangsstärke zwischen 1 und 2 *Balken*, vereinzelt werden sogar 3 *Balken* erreicht. Ist man allerdings über *GPRS* verbunden, so liegt die Empfangsstärke konstant bei 3 bis 5 *Balken*. Stabil blieb die *UMTS*-Verbindung immer in der Nähe der Fenster in Richtung Lehnerz/Fulda Stadt. In dieser Richtung wird sich der nächste UMTS-Sendemast befinden und der Empfang weniger gedämpft. Der Test wird nun zweimal durchgeführt, und zwar einmal bei der Standard-Bedingung von 1-2 Balken Empfangsstärke und bei der bestmöglichen Bedingung von 3 Balken Empfangsstärke.

Wenn eine Verbindung aufgebaut wurde und diese stabil bleibt, hat man eine gute Verbindung zum Internet, die, je nach Empfangsstärke, oftmals merklich schneller ist als eine *ISDN*-Verbindung. *Microsoft Windows* meldet, dass eine Verbindung mit 384 kbit/s hergestellt wurde. Auch die *Vodafone Mobile Connect*-Software signalisiert, dass eine Verbinung besteht und zeigt in der Statusleiste die Dauer der Verbindung und die momentane Upload- und Download-Rate an.

A.5.1.1 Standard-Bedingung – 1-2 *Balken* Empfangsstärke

Der erste Test ist das Surfen im Internet. Als erste Seite wird die Webseite von *Google Deutschland* aufgerufen. Die Seite eignet sich besonders für einen Verbindungstest, weil die Webseite ohne viel Grafiken und Animationen auskommt. Es dauert nicht lange und die Google-Webseite steht zur Verfügung. Die Logo-Grafik sieht etwas verpixelter aus, als beispielsweise bei einer DSL-Verbindung. Das liegt an der Verbindungsoptimierung der *Vodafone Mobile Connect*-Software, die eine zusätzliche Kompression einschaltet, um Bandbreite zu sparen. Die zweite Seite, die *Heise Online*-Seite, dauert schon etwas länger, weil hier mehr Grafiken und Informationen angezeigt werden müssen. Man könnte die Geschwindigkeit des Seitenaufbaus etwa mit der einer ISDN-Verbindung vergleichen. Als letzte Seite wird die Webseite der *EDAG Engineering & Design AG* aufgerufen, die schon deutlich grafiklastiger ist. Auch hier dauert wieder der Seitenaufbau etwas länger, bleibt aber im Bereich einer ISDN-Verbindung.

Um nun heraus zu bekommen, wo ungefähr die durchschnittliche Datenrate bei dieser Verbindung liegt, wird die 4 MB-Datei heruntergeladen. Auch hier wird ungefähr die Geschwindigkeit einer ISDN-Verbindung erreicht, bei einer durchschnittlichen Downloadrate von 48 kbit/s und einer Dauer von ca. 10 Minuten bis die Datei auf dem Laptop ist. Der maximale Wert liegt bei ca. 100 kbit/s. Das ist schon ein ordentlicher Wert, mit dem sich auf jeden Fall arbeiten läßt. Eine ping-Analyse der Verbindung ergibt eine durchschnittliche Antwortzeit der einzelnen Webseiten von ca. 400 ms.

Eine VPN-Verbindung zum Netzwerk der *EDAG Engineering & Design AG* ist unter diesen Bedingungen jedoch problematisch. Es dauert sehr lange, bis *Lotus Notes* eine Verbindung zum Mailserver aufbaut, manchmal kam es sogar zu einem *Timeout*. Es sind mehrere Versuche nötig, bis man in seinem Postfach ist und eine Mail geschrieben und abgeschickt hat. Der Zugriff auf das Netzwerk der *EDAG Engineering & Design AG* bei diesen Bedingungen ist also nicht zufriedenstellend.

A.5.1.2 bestmögliche Bedingung – 3 *Balken*

Da der Test in bei Standard-Bedingungen nicht sehr überzeugend war, wird nun auch die bestmögliche Bedinung getestet und es gibt einen merklichen Unterschied. In dieser neuen Bedingung hat man nicht mit ständigen Verbindungsabbrüchen zu kämpfen, die in der Standard-Bedingung immer wieder auftreten. Auch trägt dieser eine Balken mehr an Empfangsstärke eine deutliche Verbesserung der Performance bei.

Schon der Verbindungsaufbau geht viel schneller von Statten und auch die drei Webseiten werden in kurzer Zeit aufgebaut. Von der Darstellungsgeschwindigkeit befindet sich man bei einer normalen Internet-Recherche bereits im DSL-Bereich. Die Messung beim Download der 4 MB-Datei zeigt auch eine deutliche Verbesserung der Übertragungsgeschwindigkeit: die durchschnittliche Datenrate liegt bei ca. 333 kbit/s, das Maximum sogar bei 507 kbit/s, und die Datei ist in ca. 1:30 Minuten heruntergeladen. Man liegt hier also nahe an der theoretisch erreichbaren Geschwindigkeit von 384 kbit/s.

Nun bleibt es abzuwarten, ob diese Leistungen auch bei einer VPN-Verbindung erhalten bleiben. Der Aufbau der Verbindung geht deutlich schneller und auch der Zugriff auf das *Lotus Notes*-Postfach ist in einem Bruchteil einer Sekunde möglich. Problemlos läßt sich das Postfach aufrufen und eine eMail schreiben und versenden. Da diese Schnelligkeit überzeugt wird nun versucht auch auf das Home-Verzeichnis im Netzwerk zuzugreifen. Es dauert sehr lange bis das Netzwerk angezeigt wird und man Zugriff auf die Netzwerkumgebung hat. Im Test bricht der Verbindungsaufbau meist mit einem *Timeout* ab. Somit ist es nicht möglich auf das Home-Verzeichnis zuzugreifen.

A.5.2 Stadtgebiet Fulda

Im innerstädtischen Bereich ist das UMTS-Netz deutlich besser ausgebaut. Hier hat man eine konstante Empfangsstärke von 4 bis 5 Balken, ebenso wie beim GPRS-Netz. Die Verbindung zum UMTS-Netz ist dabei sehr stabil und kann problemlos benutzt werden, um im Internet zu recherchieren. *Microsoft Windows* meldet wieder eine 384 kbit/s Datenrate.

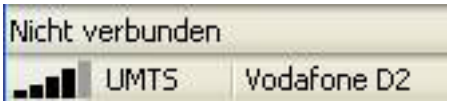


Abb. A.7: *Empfangsstärke von UMTS im Stadtgebiet Fulda*

Sogleich beginnt wieder der normale Test mit dem Aufrufen der drei Test-Webseiten. Die *Google*-Webseite ist noch schneller da, wie es im *TEC-Center* der Fall ist. Ebenso die *Heise Online*-Webseite. Die Seiten bauen sich blitzschnell auf und es sind kaum merkliche Verbindungsunterschiede zu DSL erkennbar. Auch bei ausgeschalteter Kompression bleibt die Verbindung stabil und die Webseiten bauen sich im gleichen Tempo auf. Der Download der 4 MB-Datei dauert nun nur noch etwas über eine Minute mit einer durchschnittlichen Datenrate von 380 kbit/s , das Maximum lag bei ca. 470 kbit/s). Da dies ein recht guter Wert ist, wirft sich nun die Frage auf, wie schnell man die gleiche Datei per FTP auf einen Server laden kann. Hier liegt die Dauer bei ca. 6:30 Minuten bei einer durchschnittlichen Datenrate von 55 kbit/s, mit einem Maximum von 120 kbit/s. Hier befinden wir uns an den Grenzen der theoretischen Möglichkeit und evt. darauf zurückzuführen, dass die Teilnehmerzahl im *UMTS*-Netz noch sehr gering ist.

Die VPN-Verbindung zum Netzwerk der *EDAG Engineering & Design AG* ist ebenfalls in sehr kurzer Zeit aufgebaut. Der Zugriff auf das *Lotus Notes*-Postfach geht innerhalb von wenigen Augenblicken ab und die eMail ist ohne Probleme geschrieben und gesendet. Der Zugriff auf das Home-Verzeichnis im Netzwerk der *EDAG Engineering & Design AG* stellt nun kein Problem mehr dar.

A.5.3 Randgebiet Fulda

Beim letzten Testszenario ist es nun die Frage, wie weit *UMTS* überhaupt in Fulda zur Verfügung steht und in wiefern sich eine Anschaffung zum heutigen Zeitpunkt lohnt. Die ersten beiden Testszenarien haben schon deutlich gezeigt, welche Einbußen man mit niedrigeren Empfangsstärken hingenommen werden müssen.

Dieser Test wurde mittels eines *UMTS War-Drives* druchgeführt. Dazu wurde der Laptop mit der eingesteckten *Mobile Connect Card UMTS* in ein Auto gestellt und während der Fahrt die Empfangsstärke über die Anzeige in der *Vodafone Mobile Connect*-Software gemessen. Dieser Test ist wieder nicht als wissenschaftlich anerkannt zu betrachten, aber die Anzeige in der *Vodafone Mobile Connect*-Software gibt eine gute Einschätzung über die Verfügbarkeit von *UMTS* in Fulda. Als Route wurde die Stadt Fulda einmal komplett umfahren. Ausgangspunkt ist der *TEC-Center* der *EDAG Engineering & Design AG* in der Steinauer Strasse in Petersberg.

Wenn man sich weiter Richtung Steinau bewegt, bricht recht schnell die UMTS-Verbindung ab und die *Mobile Connect Card UMTS* sucht ein neues Netzwerk. Weiter geht es in Richtung Petersberg, wo man durchschnittlich 2 bis 3 *Balken* Empfang hat, im Industriegebiet Petersberg/Künzell sogar vollen Ausschlag mit 5 *Balken*. Weiter Richtung Künzell durch die Wohngebiete hat man einen Empfang von 2 bis 3 *Balken*, was bis zur *Rhön-Therme* auf 5 *Balken* steigt. Kurz hinter der *Rhön-Therme* befindet sich ein großer Funkmast mit UMTS-Funkantennen. Pilgerzell und Engelhelms sind wieder komplett außerhalb des Versorgungsgebietes. Beim *Schloss Adolphseck* erhöht sich der Empfang wieder auf 2 bis 3 *Balken*, was sich bis nach Bronnzell hält.

Nun geht es über die Frankfurter Straße quer durch Fulda mit 3 bis 5 *Balken* Empfang. Auffällig ist wieder der volle Ausschlag (5 *Balken*) im Industriegebiet *Kohlhäuser Feld*. Auf der Bardostraße bleibt der gute Empfang von 4 bis 5 Balken. Auch in Richtung des neuen Stadtteils *Fulda Galerie* und Sickels bleibt es bei dieser Empfangsstärke. Grund hierfür ist der ehemalige Tower auf dem Sickelser *Airfield*, einem Tower eines ehemaligen Flugplatzes der US Armee. Dort sind zahlreiche *UMTS*-Antennen installiert, die bis nach Heimbach strahlen. Dort erreicht man ebenfalls 4 bis 5 *Balken* an Empfangsstärke. In Maberzell bricht diese Verbindung allerdings wieder ab. Nur teilweise hat man eine Empfangsstärke von maximal 1 *Balken*. Zurück Richtung Stadt Fulda steigt der Empfang im Stadtteil Horas wieder auf 2 bis 3 *Balken*, was auch für Niesig gilt. Endstation ist das Industriegebiet *Eisweiher* und die Fachhochschule Fulda. Dort hat man wieder vollen Ausschlag, also 5 *Balken* Empfangsstärke.

Allgemein ist zu sagen, dass vor allem in den Industriegebieten und den neuen Stadtteilen das UMTS-Netz schon sehr gut ausgebaut ist. Auch ist die Höhe des Standortes sehr ausschlaggebend. So hat man in einem Stadtteil plötzlich einen besseren Empfang, wenn man sich auf einer Anhöhe befindet, als im restlichen Stadtteil. Als Beispiel wäre der Rauschenberg in Petersberg mit 4 *Balken* Empfang zu nennen als im restlichen Petersberg mit nur 2 bis 3 *Balken*. Stadtteile die komplett in einer Senke liegen, werden meist gar nicht vom *UMTS*-Netz erreicht. Auch wenn man sich weiter stadtauswärts bewegt, sinkt die Empfangsstärke sehr schnell. Bis auch in der Rhön ein zufriedenstellend *UMTS*-Empfang möglich ist, wird es wohl noch einige Zeit dauern.

A.6 Fazit

Allgemein betrachtet ist *UMTS* eine sehr interessante Technik mit der viele Sachen möglich sind. Vor allem lange Wartezeiten beim mobilen Surfen gehören der Vergangenheit an, vorausgesetzt man hat einen guten und stabilen UMTS-Empfang. Sind diese Voraussetzungen gegeben, merkt man kaum einen Unterschied zu heutiger DSL-Technik und -Geschwindigkeiten. Allerdings läßt der Ausbau des UMTS-Netzes noch sehr zu wünschen übrig, zumindest für die Region Fulda. In großen Ballungsgebieten, wie z.B. das Rhein-Main-Gebiet, mag die Ausbaustufe schon fortgeschrittener sein und nicht mehr so viele Kinderkrankheiten haben, wie hier in Osthessen. Die Frage bleibt noch, wie sich das Netzwerk verhält, wenn die Anzahl der Teilnehmer steigt, die das Netz für die Datenkommunikation benutzen wollen. Da die Bandbreite nicht exklusiv für jeden zur Verfügung gestellt wird, sondern ähnlich wie bei GSM, unter allen Teilnehmern in der gleichen Zelle aufgeteilt wird, wird auch sehr schnell die Datenrate sinken. Momentan sind noch nicht viele Teilnehmer im UMTS-Netz, aber in einem halben Jahr sieht es evt. schon anders aus. Vielleicht ist zu diesem Zeitpunkt ein erneuter Test sinnvoll, um auch diese Probleme zu analysieren. Ebenso ist zusätzlich noch zu teste, in wiefern eine extern an die *Mobile Connect Card UMTS* angeschlossene Antenne die Empfangsleistung erhöht.

Die *Mobile Connect Card UMTS* ist einfach zu installieren und einfach in der Bedienung. Die Installationsanweisungen sind recht selbsterklärend und bedürfen keinem großen technischen Verständnis. Somit ist die Installation auch für EDV-Laien einfach durchzuführen. Inwiefern mal eine Unterstützung für Linux-Systeme geplant ist, bleibt abzuwarten. Da die *Mobile Connect Card UMTS* aber als einfache Modem-Karte erkannt wird und man eine Verbindung über DFÜ aufbauen kann, so dürfte sie auch unter Linux nutzbar sein. Somit ist die *Mobile Connect Card UMTS* eine gute Wahl für den Datenaustausch im UMTS-Netzwerk.

Die *Vodafone Mobile Connect*-Software ist ein interessantes Tool, das eine gute Verwaltung der SIM-Daten und eine einfache Benutzung der Möglichkeiten der UMTS-Kommunikation bietet. Die Software ist ebenfalls sehr einfach zu bedienen. Die Dialoge sind leicht und verständlich und bieten kaum Verständnisschwierigkeiten. Die Online-Hilfe ist ebenfalls sehr gut aufgebaut und bietet zusätzlich eine sehr gute Unterstützung, falls doch mal eine Funktion nicht verstanden wurde. Jedoch fehlt im Bereich *Nutzung* eine Übersicht der insgesamt verbrauchten Zeit. Es werden zwar die übertragenen Datenmengen, getrennt nach GPRS und UMTS angezeigt, aber kein Hinweis auf die dafür benötigte Zeit. Dies ist bei einem zeitorientierten Tarif nicht sehr hilfreich und man verliert leicht die Kontrolle. Bei einem datenorientierten Tarif gibt dieser Bereich aber eine gute Übersicht über die übertragenen Datenmengen. Es ist auf jeden Fall zu empfehlen die neue Version 3.0.2 zu installieren. Diese Version enthält weniger Fehler als die Software, die mit der *Mobile Connect Card UMTS* mitgeliefert wurde.

Letztendlich bleibt zu sagen, dass sich der Einsatz zum heutigen Zeitpunkt zumindest für den Großraum Fulda eher uninteressant ist. Man hat oft mit Verbindungsschwierigkeiten zu kämpfen, die oft Zeit und Nerven, aber auch Geld kosten. Vor allem wenn man die mobile Freiheit nutzen will, ist der Einsatz eher nicht empfehlenswert. Ist man allerdings ständig an einem gut ausgebauten Ort (3 *Balken* oder höher), dann ist es bestimmt eine Alternative zum kabelgebundenen DSL-Anschluß.

Aber auch der Preis sollte nicht außer Acht gelassen werden. Die Anschaffung der Karte ist noch relativ gering. Bei einem gleichzeitigen Abschluß eines Mobilfunkvertrages bei *Vodafone* bekommt man die *Mobile Connect Card UMTS* schon ab 1,- €, ohne diesen Vertrag schlägt die *Mobile Connect Card UMTS*mit ca. 500,- € zu Buche. Das teure sind dann aber die Datentarife von Vodafone. Hier wird zunächst zwischen zeitgebundenen und volumengebundenen Tarifen unterschieden. Die zeitgebundenen Tarife kosten in der Grundgebühr von 9,86 € inkl. 2 Std. bis hin zu 69,60 € inkl. 30 Std. Jede weiteren 10 Minuten kosten dann entsprechend von 1,86 € bis hin zu 0,93 €. Die volumengebundenen Tarife sind von den Kosten her die gleichen wie die zeitgebundenen Tarife, jedoch sind dort von 10 MB bis 150 MB inkl. und ab dann wird je angefangenes MB abgerechnet[12].

Die Technik ist noch im Aufbau und die Versorgungsgebiete wachsen stetig und schnell. In einem halben Jahr oder Jahr dürfte *UMTS* flächendeckender in Deutschland verfügbar und somit die neue mobile Freiheit gut nutzbar sein. Auch ist die Kostenentwicklung weiter zu beobachten. Alleine Vodafone hat während seinem offiziellen UMTS-Start seine Preise zweimal reduziert. Wenn das UMTS-Netz auf keine große Akzeptanz bei den Nutzern stößt, werden die Preise vermutlich noch weiter sinken. Zum momentanen Zeitpunkt finde ich den Einsatz von UMTS noch nicht für sinnvoll, weil das System noch sehr unter Problemen leidet. Diese werden aber in der nächsten Zeit immer weniger und in einem halben Jahr, Jahr dürfte der Einsatz von UMTS bestimmt eine Bereicherung für jedes Unternehmen sein.

Weitere Informationen und Tests zu der *Mobile Connect Card UMTS*gibt es unter:

1. UMTSlink.at, Meinungen und Tipps zur *Mobile Connect Card UMTS*
   `http://umtslink.at/cgi-bin/reframer.cgi?../telefone/3g/vodafone_data card.php`
2. 3G-Mobilfunkforum von UMTSlink.at
   `http://www.umtslink.at/3g-forum/showtopic.php?threadid=2154`
3. tecChannel, Erster UMTS-Test: Vodafone UMTS-PC-Card
   `http://www.tecchannel.de/internet/1350/`
4. ZDF Online, Handyloser UMTS-Start bei Vodafone (inkl. Video)
   `http://www.heute.t-online.de/ZDFheute/artikel/31/0,1367,COMP-0-2104031 ,00.html`
5. Linux and the Vodafone Mobile Connect (UMTS) 3G/GPRS Datacard
   `http://kuix.de/umts/vodafone/`

[12]Die Preise und Konditionen entsprechen einem Stand vom August 2004. Aktuelle Preise sind auf der Webseite von *Vodafone* nachzulesen.

A.7 Anhänge

A.7.1 Installationsanleitung für Vodafone Dashboard - Version 3.0.2

Systemanforderungen
Datenkarte: Mobile Connect Card GPRS, Mobile Connect Card UMTS oder Mobile Connect Card W-LAN (SonyEricsson GC79) Betriebssystem: Windows XP, XP Pro, 2000 (Service Pack 2 und höher), 98 (SE), ME, NT 4 (SP 6a)

Installationsanleitung
So können Sie die neue Dashboard-Version ganz einfach auf Ihrem Notebook installieren:

Hinweis: Installieren Sie bitte die Software vor dem Einstecken der Mobile Connect Card ins Notebook Deinstallieren Sie zunächst ggfs. eine ältere Version des Vodafone Dashboards.

1. Entpacken Sie die ZIP-Datei und starten Sie den Installationsvorgang.
2. Wählen Sie Land und Sprache aus: Die Sprache des Betriebssystems ist automatisch eingestellt. Als Land wählen Sie das Land Ihres heimischen Netzbetreibers aus, in diesem Fall Deutschland für Vodafone D2. Unter dem Menüpunk „Hilfe“ erhalten Sie dazu weitere Hinweise.
3. Anschließend überprüft das Installationsprogramm, ob Ihr Betriebssystem den Mindestanforderung entspricht.
4. Lesen Sie den Lizenzvertrag durch und klicken Sie auf „Akzeptieren“.
5. Wählen Sie nun den „Installationstyp“ aus. Es gibt folgende Optionen:
   - Internetzugang
   - Zugang zum Firmennetz
   - Benutzerdefinierter Zugang
6. Wenn Sie den Installationstyp „Zugang zum Firmennetz“ wählen, können Sie ein VPN-Programm nutzen. Bei Fragen zu Ihrem Firmennetz wenden Sie sich bitte an Ihren zuständigen IT-Administrator.
7. Wennn Sie den Installationstyp „Internetzugang“ wählen, können Sie zusätzlich ein Instant-Messenger-Progamm (IM) nutzen. Wählen Sie nach Bedarf Ihr IM-Programm aus.
8. Die von Ihnen gewählten Optionen werden in der „Übersicht“ angezeigt. Falls Sie noch Änderungen vornehmen möchten, klicken Sie auf „Zurück“, ansonsten klicken Sie auf „Weiter“, um die Installation fortzusetzen.
9. Der InstallShield Wizard führt Sie nun weiter durch den Installationsprozeß.
10. Geben Sie unter „Benutzerinformationen“ die Benutzer ein, die die Mobile Connect Card-Software benutzen dürfen. Sie können einen bestimmten oder mehrere Nutzer anlegen. Klicken Sie danach auf „Weiter“.

11. Wählen Sie jetzt den „Setuptyp“ aus. Sie haben Wahl zwischen den Optionen „Vollständig“ (empfohlen) oder „Angepaßt“ (für erfahrene Benutzer). Klicken Sie danach auf „Weiter“.
12. Starten Sie nun die eigentliche Installation der Software mit „Installieren“. Die Software wird nun installiert. Das kann einige Minuten dauern. Sobald die Meldung „InstallShield Wizard abgeschlossen“ erscheint, wurde die Software erfolgreich installiert.
13. Beenden Sie den Installationsvorgang, indem Sie auf „Fertigstellen“ klicken.
14. Jetzt müssen Sie Ihren Computer neu starten.
15. Nach dem Neustart wird in der Regel der „Profilmanager“ angezeigt. Wird der „Profilmanager“ nicht angezeigt, müssen Sie das Programm „Vodafone Mobile Connect“ (VMConnect.exe) starten. Sie finden das Programm in der Regel im Verzeichnis `C:\Programme\Vodafone\Vodafone Mobile Connect`. Starten Sie das Programm mit einem Doppelklick auf „VMConnect.exe“.
16. Sie müssen jetzt im „Profilmanager“ ein Profil anlegen. Wählen Sie „PC-Datenkarte“ und klicken Sie auf „Weiter“.
17. Wählen Sie nun die passenden Optionen für Ihre Datenkarte aus (Hersteller, Modell, Dienste-Typ).
18. Das Programm fordert Sie nun auf, die Vodafone Mobile Connect Card in den dafür vorgesehenen PC-Karten-Steckplatz einzustecken. Sie finden diesen Steckplatz (PCMCIA-Steckplatz) entweder an der Seite oder an der Rückseite Ihres Notebooks.
19. Bitte warten Sie, bis das Betriebssystem Ihres Rechners die Datenkarte erkannt und die erforderlichen Treiber installiert hat. Das kann ungefähr eine Minute dauern.
20. Wählen Sie dann im Fenster „Profil erstellen“ ein E-Mail-Programm aus und klicken Sie auf „Weiter“.
21. Überprüfen Sie Ihre Auswahl und klicken Sie auf „Fertigstellen“, wenn alles korrekt ist. Möchten Sie noch etwas ändern, klicken Sie auf „Zurück“ und nehmen Ihre Änderungen vor.
22. Fertig. Ihre Vodafone Mobile Connect Card ist nun einsatzbereit.

Quelle: http://www.vodafone.de/business/support/49943.html

A.7.2 Traceroute UMTS EDAG

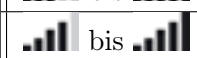
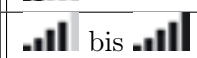
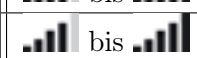
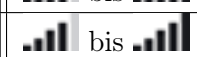
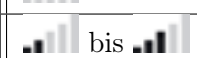
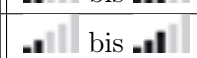
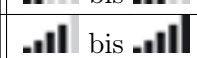
```
Routenverfolgung zu ica.edag.de [194.76.195.141]  Über maximal 30 Abschnitte:

  1    357 ms    429 ms    449 ms  10.242.176.1
  2    346 ms    400 ms    439 ms  172.23.9.17
  3    334 ms    389 ms    399 ms  172.17.6.17
  4    342 ms    398 ms    419 ms  172.17.0.9
  5    338 ms    408 ms    419 ms  172.17.0.94
  6    349 ms    419 ms    409 ms  172.17.12.2
  7    424 ms    419 ms    429 ms  10.210.0.244
  8    356 ms    409 ms    408 ms  10.210.0.250
  9    338 ms    420 ms    439 ms  139.7.127.1
 10    356 ms    408 ms    400 ms  139.7.127.254
 11    343 ms    439 ms    419 ms  ffm-b2-pos4-2.telia.net [213.248.79.149]
 12    420 ms    429 ms    419 ms  ffm-bb2-pos2-3-0.telia.net [213.248.64.177]
 13    377 ms    428 ms    449 ms  prs-bb2-pos7-0-0.telia.net [213.248.65.117]
 14    384 ms    470 ms    468 ms  ldn-bb2-pos7-0-0.telia.net [213.248.65.113]
 15    393 ms    479 ms    479 ms  ldn-b1-pos5-0.telia.net [213.248.74.14]
 16    397 ms    531 ms    459 ms  linx-gw2.uk.psi.net [195.66.226.14]
 17    586 ms    509 ms    439 ms  t1-2.LDN2.psie.net [154.14.65.11]
 18   1302 ms    549 ms   1629 ms  t1-1.FRA5.psie.net [154.14.65.29]
 19    438 ms    460 ms    470 ms  t1-1.MUC2.psie.net [154.14.65.8]
 20    396 ms    479 ms    468 ms  154.14.70.46
 21    402 ms    469 ms    479 ms  154.14.155.57
 22    430 ms    489 ms    509 ms  194.76.195.1
 23    417 ms    443 ms    461 ms  ica.edag.de [194.76.195.141]

Ablaufverfolgung beendet.
```

A.7.3 UMTS-Wardrive Fulda

Die folgende Liste ist eine subjektive Erfassung der Erreichbarkeit des UMTS-Netzes in Fulda. Wer genauere Daten haben möchte wendet sich am Besten an die jeweiligen Netzbetreiber.

Stadtgebiet	**Empfangsstärke**
EDAG TEC-Center	
Steinau	
Petersberg	
Industriegebiet Petersberg/Künzell	
Künzell	
Rhön-Therme	
Pilgerzell	
Engelhelms	
Schloss Fasanerie	
Bronnzell	
Industriegebiet *Kohlhäuser Feld*	
Fulda, Frankfurter Straße	
Fulda, Bardostraße	
Johannesberg	
Fulda Galerie	
Sickels	
Haimbach	
Maberzell	
Horas	
Niesig	
Industriegebiet *Eisweiher*	
Fachhochschule Fulda