Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Diplomarbeit

Entwurf einer Sicherheitsarchitektur für den Einsatz mobiler Endgeräte

vorgelegt von
Andre Lüpke
am 20.04.2004

Studiengang Softwaretechnik

Betreuender Prüfer: Prof. Dr.-Ing. Martin Hübner
Koreferent: Prof. Dr. Kai von Luck

Fachbereich Elektrotechnik und Informatik
Department of Electrical Engineering and Computer Science

André Lüpke

Entwurf einer Sicherheitsarchitektur für den Einsatz mobiler Endgeräte

Diplomarbeit eingereicht im Rahmen der Diplomprüfung
im Studiengang Softwaretechnik

am Fachbereich Elektrotechnik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Martin Hübner
Zweitgutachter: Prof. Dr. Kai von Luck

Abgegeben am 20. April 2004

André Lüpke

Thema der Diplomarbeit

Entwurf einer Sicherheitsarchitektur für den Einsatz mobiler Endgeräte

Stichworte

PDA, mobile Endgeräte, Sicherheit, Sicherheitsarchitektur

Kurzzusammenfassung

In Zukunft werden immer mehr mobile Endgeräte für die Durchführung geschäftlicher Transaktionen genutzt werden. Ein Hauptproblem bilden dabei die bisher wenig beachteten Sicherheitsrisiken mobiler Endgeräte.
In dieser Diplomarbeit wird ausgehend von einem Beispielszenario eine Sicherheitsarchitektur entworfen, welche die gefahrlose Nutzung von Diensten gewährleistet. Dazu wird ein Maßnahmenkatalog erarbeitet und eine Anwendung zur sicheren Inanspruchnahme von Diensten konzipiert. Diese Anwendung wird anschließend prototypisch implementiert.

André Lüpke

Title of paper

Design of a security architecture for the usage of mobile devices

Keywords

PDA, mobile devices, security, security architecture

Abstract

In the near future more and more mobile devices will be used to perform business dealings. On the other hand, there hasn’t been paid much attention to the related security risks.
Based on a scenario a security architecture will be presented. A package of measures will be constructed and an application for using services will be designed. As a prove of concept a prototypical implementation concludes this thesis.

Danksagung

Ich möchte mich hiermit bei Herrn Prof. Dr.-Ing. Martin Hübner und Herrn Prof. Dr. Kai von Luck für ihre Unterstützung bedanken. Sie standen mir immer mit Rat und Tat zur Seite und ich hätte mir keine besseren Betreuer vorstellen können. Außerdem möchte ich mich bei meiner Freundin für ihre unendliche Geduld bedanken. Ein weiterer Dank geht an alle diejenigen, die mich in irgendeiner Art unterstützt haben, insbesondere Piotr Wendt, Michael Knop und Björn Jensen. Abschließend bedanke ich mich bei meinen Eltern, ohne deren Unterstützung ich dieses Studium niemals hätte machen können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Wireless e-business can't go to air without Security

IBM

1.1 Motivation

In einer Zeit, in der das Wort Mobilität immer mehr an Bedeutung gewinnt, wächst der Bedarf an sicheren Geräten, die diesen Trend sinnvoll unterstützen. Es existieren bereits viele Geräte, wie z.B. PDAs (Personal Digital Assistants), die es Benutzern ermöglichen, jederzeit und überall auf die verschiedensten Informationen und Dienste zuzugreifen. Oftmals wird dabei jedoch das Thema Sicherheit vernachlässigt oder gänzlich außer Acht gelassen. In dieser Arbeit sollen deshalb mobile Endgeräte anhand eines Beispielszenarios auf ihre Sicherheit untersucht werden und daraufhin eine Sicherheitsarchitektur entworfen werden, die insbesondere die Nutzung von Diensten in internen Netzen berücksichtigt.

1.2 Zielsetzung

Ziel ist es, mögliche Bedrohungen des gewählten Szenarios zu analysieren und anschließend eine sichere Architektur für den Einsatz mobiler Endgeräten zu entwerfen. Als Beispielszenario dient in diesem Fall ein Ferienclub, welcher seinen Gästen PDAs aushändigt. Über diese Geräte sind die Urlauber in der Lage vom Club angebotene Dienste zu nutzen. Einen guten Überblick über weitere denkbare Szenarios bietet (IZT u. a., 2001). Die Sicherheitsarchitektur soll dabei aus einem Maßnahmenkatalog zur Absicherung des Szenarios und dem Entwurf einer Software zur sicheren Nutzung von Diensten bestehen. Diese Software soll daraufhin prototypisch implementiert werden. Als mobile Endgeräte werden in dieser Arbeit ausschließlich PDAs untersucht. Es sollte jedoch ohne weiteres möglich sein, den Entwurf auf andere mobile Endgeräte zu übertragen. Aus diesem Grund wird in dieser Arbeit des Öfteren der Begriff „PDA“ stellvertretend für „mobiles Endgerät“ verwendet.

1.3 Gliederung der Arbeit

Der Themenkomplex der Arbeit wird anhand eines Beispielszenarios im Kapitel 2 vorgestellt. Es wird erläutert, in welchem Kontext diese Arbeit steht und es werden bereits erste Anforderungen an das System skizziert.

In Kapitel 3 erfolgt daraufhin eine detaillierte Analyse der Anforderungen an das System mit Hilfe von Anwendungsfalldiagrammen. Dafür werden die wichtigsten Anwendungsfälle und Rollen herausgearbeitet und die relevanten Rollen auf ihre möglichen Handlungen innerhalb des Systems untersucht. Des Weiteren werden allgemeine Anforderungen aus Sicht der Informatik definiert, welche das System ebenfalls erfüllen soll.

Die besonderen Sicherheitsaspekte, die es bei dem System zu berücksichtigen gilt, werden in Kapitel 4 aufgeführt. Dazu werden zuerst alle relevanten Daten in ihrer Wichtigkeit und ihrem Schutzbedürfnis bewertet. In der sich anschließenden Bedrohungsanalyse wird dann auf Gefahren eingegangen, denen ein solches System ausgesetzt ist.

Aufbauend auf den gefundenen Bedrohungen findet in Kapitel 5 der Entwurf einer einheitlichen Sicherheitsarchitektur statt. Dabei werden generelle Sicherheitsmaßnahmen und der Entwurf einer Software zur sicheren Nutzung von Diensten beschrieben.

Das darauf folgende Kapitel 6 beschäftigt sich damit, wie das entworfene Gesamtsystem aus Kapitel 5 unter den Laborbedingungen prototypisch umgesetzt wurde.

In Kapitel 7 werden die Ergebnisse der Arbeit noch einmal zusammengefasst. Des Weiteren wird ein Ausblick darauf gegeben, wie sich das entworfene System erweitern ließe und was die Zukunft bringen wird bzw. bringen könnte.

2 Beispielszenario

In dem folgenden Beispielszenario geht es um einen fiktiven Ferienclub. Dieser möchte seinen Gästen die Möglichkeit bieten, bestimmte Dienste über ein geliehenes mobiles Endgerät ortsunabhängig zu nutzen. Die Gäste des Clubs erhalten dazu beim Einchecken einen PDA, über den sie diese Dienste in Anspruch nehmen können. Das Szenario wird aus Sicht des Urlaubers, Herrn Meier, beschrieben.

In dem Beispielszenario werden aufgrund des Umfangs nicht alle Vorgänge beschrieben, die in einer solchen Umgebung denkbar wären. Eine Aufschlüsselung aller Anwendungsfälle und weitere Anforderungen an das Gesamtsystem werden im Kapitel 3 beschrieben.

2.1 Reise buchen

Bevor Herr Meier seinen Urlaub in dem oben genannten Club antreten kann, führt er eine Buchung bei einem Reisebüro durch. Das Reisebüro übermittelt daraufhin die Daten von Herrn Meier an den Ferienclub. Der Club ist fortan über den Ankunftstag und die Aufenthaltsdauer des Gastes informiert. Er kann sich daher bereits frühzeitig auf die Ankunft von Herrn Meier vorbereiten.

2.2 Einchecken

Nachdem Herr Meier in seinem Urlaubsland angekommen ist, begibt er sich als Erstes zur Rezeption des Ferienclubs, um einzuchecken. Er weist seine Identität nach und bekommt daraufhin seine Zimmerschlüssel ausgehändigt. Nachdem Herrn Meier der Schlüssel übergeben wurde, informiert ihn der Portier darüber, dass er einen PDA vom Ferienclub gestellt bekommen kann. Mit diesem wäre er dann in der Lage, verschiedene urlaubsbezogene Angebote auf einfach Weise ortsunabhängig zu nutzen.

Herr Meier entscheidet sich für die Nutzung des Gerätes. Er bekommt von dem Portier eine kurze Einweisung und nachdem ein paar letzte Anpassungen vorgenommen wurden, wird ihm das Gerät zusammen mit den für die Authentifizierung notwendigen Daten ausgehändigt. Diese so genannten Authentifizierungsdaten werden benötigt, damit sich Herr Meier

an dem Gerät anmelden und daraufhin die Dienste des Clubs über den PDA in Anspruch nehmen kann.

Abschließend wird Herrn Meier mitgeteilt, dass er sich bei weiteren Fragen wieder an der Rezeption melden kann.

In seinem Zimmer angekommen, schaltet Herr Meier das Gerät ein, um sich über die bereitgestellten Dienste zu informieren. Nach einer erfolgreichen Authentifizierung am Gerät steht ihm eine Hilfefunktion zur Verfügung. Diese zeigt ihm, auf welche Dienste er über das Gerät zugreifen kann.

2.3 Allgemeine Dienste nutzen

Herr Meier ist mit dem PDA in der Lage sich jederzeit über Neuigkeiten und Angebote des Ferienclubs zu informieren.

Unter den Nachrichten, die an die Clubgäste versendet werden, befinden sich u.a. Hinweise über das Unterhaltungsprogramm des Clubs, über Sport- und Reiseveranstaltungen, sowie über Veranstaltungen in der näheren Umgebung.

Eine beispielhafte Gliederung der Rubriken könnte wie folgt aussehen:

1. Sport
    a) Wandern
    b) Fußball
    c) Volleyball
    d) Klettern
    e) Surfen
2. Spiele
    a) Kartenspiele
    b) Brettspiele
3. Nachrichten
    a) Wirtschaft
    b) Börse
    c) Lokales
4. Veranstaltungen

a) Ausflüge

b) Gewinnspiele

c) Parties

d) Konzerte

Ein weiterer Dienst bietet Herrn Meier die Möglichkeit, Reservierungen für bestimmte Einrichtungen vorzunehmen. Er kann somit unter anderem den Tennisplatz des Clubs oder einen Platz im Restaurant reservieren.

Die bisher beschriebenen Dienste benötigten alle eine Verbindung zum drahtlosen Clubnetzwerk. Es gibt jedoch auch Dienste, die sich direkt auf dem Gerät befinden und auch außerhalb des Clubgeländes funktionieren. Ein integrierter Reiseführer steht Herrn Meier beispielsweise auch dann zur Verfügung, wenn er sich nicht auf dem Clubgelände aufhält. Des Weiteren gibt es Spiele und die in allen PDAs vorhandenen Programme zur Terminverwaltung, Adressverwaltung etc., welche allgemein als PIM (Personal Information Management) Funktionen bezeichnet werden (siehe Abbildung 2.1).

Abbildung 2.1: Beispiel: PIM-Anwendungen des Pocket PCs

2.4 Kritische Dienste nutzen

Neben diesen Diensten, die hauptsächlich Informationscharakter haben, kann Herr Meier auch Leistungen in Anspruch nehmen, welche eine Rechtsverbindlichkeit nach sich ziehen. Typischerweise sind dies geschäftliche Transaktionen, wie z.B. Käufe. In dieser Arbeit werden solche Dienste als „kritische Dienste“ bezeichnet. Zusätzlich werden die Dienste als kritisch bezeichnet, bei denen persönliche Daten von Herrn Meier an eine dritte Person übertragen werden. An einigen Beispielen sollen solche Dienste vorgestellt werden.

Wie bereits in Kapitel 2.3 beschrieben, hat Herr Meier die Möglichkeit, sich z.B. über anstehende Ausflüge oder Konzerte zu informieren. Er ist zusätzlich in der Lage, über den PDA für diese Veranstaltungen Karten zu erwerben.

Sollte sich Herr Meier gerade am Strand befinden, so kann er über den PDA Speisen und Getränke bestellen, die ihm dann direkt an seinen Liegeplatz gebracht werden. Dazu wird die Position seines PDAs ermittelt, so dass das Personal weiß, wohin die bestellten Sachen geliefert werden müssen.

Zusätzlich kann sich Herr Meier Lose für die im Club stattfindenden Lotterien und Tombolas kaufen.

Um einen Überblick über die bisher entstandenen Kosten zu erhalten, hat Herr Meier jederzeit die Möglichkeit, sich eine Übersicht über alle getätigten Zahlungsvorgänge anzeigen zu lassen. Die Liste enthält Dienstnamen, Zeitpunkt der Inanspruchnahme und Höhe der entstandenen Kosten.

2.5 Supportleistungen des Ferienclubs

Sollte Herr Meier einmal Probleme mit seinem PDA haben, so kann er sich an der Rezeption melden. Dort wird ihm entweder sofort geholfen, oder seine Frage wird an einen vom Club angestellten PDA-Betreuer weitergeleitet. Dieser ist für die Betreuung und Verwaltung der einzelnen mobilen Endgeräte auf dem Clubgelände zuständig. Er ist ebenfalls für die Anpassungen der Geräte zuständig, die nicht an der Rezeption vorgenommen werden können. Dies umfasst zum Beispiel das Installieren zusätzlicher Hard- und Software auf dem mobilen Endgerät.

Die Rezeption ist ebenfalls Anlaufstelle, falls Herrn Meier der PDA verloren geht. In einem solchen Fall können die Mitarbeiter des Clubs an der Rezeption das Gerät als gestohlen markieren, woraufhin der Account von Herrn Meier gesperrt wird. Herr Meier muss sich somit keine Sorgen darüber machen, dass irgendjemand das abhanden gekommene Gerät missbrauchen könnte, um unter seiner Identität Dienste zu nutzen. Außerdem wird für Herrn Meier ein neues Gerät vorbereitet, welches er nach einer gewissen Wartezeit an der Rezeption abholen kann. Dabei erhält er zusätzlich zu dem Gerät neue Daten zur Authentifizierung.

2.6 Auschecken

Beim Auschecken muss Herr Meier den PDA an der Rezeption abgeben. Das Gerät wird dann gesperrt und an einen PDA-Betreuer weitergeleitet, der das Gerät daraufhin für den

nächsten Urlauber vorbereiten kann. Außerdem muss der Gast die durch die Nutzung kritischer Dienste entstandenen Kosten begleichen.

3 Anforderungsanalyse

In diesem Kapitel werden die speziellen Anforderungen erarbeitet, die sich aus dem oben beschriebenen Beispielszenario ergeben. Dafür wird zuerst eine Übersicht über die wichtigsten Anwendungsfälle und Rollen gegeben. Des Weiteren werden Rollen vorgestellt, die nicht explizit im vorangegangenen Kapitel erwähnt wurden, welche aber trotzdem für das gewählte Szenario sinnvoll oder essenziell sind.

3.1 Übersicht

Aus dem Beispielszenario lassen sich erste Handlungsabläufe und Zuständigkeiten erkennen. Diese Zuständigkeiten können einzelnen Rollen zugeordnet werden. Sinnvoll sind in diesem Zusammenhang die Rollen Gast, Rezeption und PDA-Betreuer. Abbildung 3.1 zeigt, in welchen Umgebungen die einzelnen Rollen agieren und welches die grundlegenden Zuständigkeiten bzw. Aktionen sind.

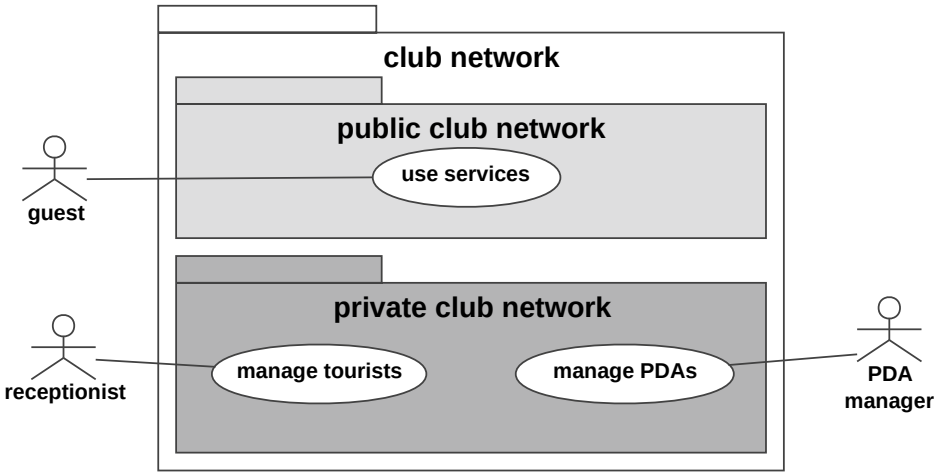

Abbildung 3.1: Anwendungsfalldiagramm: Überblick

Der Gast bzw. Urlauber (Guest) nimmt die vom Ferienclub angebotenen Dienste in Anspruch. Die Rezeption (Reception) ist für die Verwaltung und Unterstützung der Urlauber

zuständig. Alle technischen Belange, in Bezug auf das mobile Endgerät, werden vom PDA-Betreuer (PDA-Manager) gehandhabt.

3.1.1 Weitere Rollen

In dem Szenario sind noch weitere Rollen denkbar, die nicht explizit erwähnt wurden, welche aber trotzdem sinnvoll oder zwingend notwendig sind. Diese Rollen sollen hier kurz vorgestellt werden.

Reisebüro

Das Reisebüro wird in diesem System aufgeführt, da es in der Lage sein muss, den Ferienclub darüber zu informieren, dass ein Kunde eine Reise gebucht hat. Dabei sind in diesem Fall besonders der Name des Urlaubers, Anreisezeitpunkt sowie die Aufenthaltsdauer interessant. Mit diesen Daten ist der Ferienclub daraufhin in der Lage, sich auf das Eintreffen des Gastes vorzubereiten. Es kann bereits ein Account für den Gast angelegt und ein PDA vorbereitet werden.

Dienstanbieter

Die Hauptaufgabe des Dienstanbieters ist es, Leistungen zur Verfügung zu stellen, die der Gast über das mobile Endgerät nutzen kann. Dazu muss der Dienstanbieter die Möglichkeit haben, neue Dienste zu erstellen und, falls es sich um einen kostenpflichtigen Dienst handelt, sicherzustellen, dass die anfallenden Gebühren gezahlt werden.

Eine wichtige Unterscheidung in diesem Zusammenhang ist, ob es sich bei dem Anbieter um den Club selbst oder um einen Drittanbieter handelt. Der Dienst, einen Tisch im hoteleigenen Restaurant zu reservieren, unterscheidet sich somit von dem Dienst, eine Reise eines Drittanbieters über das mobile Endgerät zu buchen. Im Rahmen dieser Arbeit wird diese Unterscheidung jedoch vernachlässigt, da davon ausgegangen wird, dass die Abrechnung mit dem Drittanbieter über ein eigenes internes System erfolgt.

Administrator

Der Administrator ist für die Wartung und Verwaltung des gesamten Systems zuständig. Er muss die Möglichkeit haben alle im Netz zur Verfügung stehenden Dienste zu verwalten und zu administrieren. Er muss neue Dienste einspielen oder alte modifizieren bzw. löschen können. Er verwaltet außerdem die Rechte, die an die einzelnen Rollen, zur Nutzung von Diensten, vergeben werden können.

Zusätzlich ist er für die Sicherheit des Gesamtsystems verantwortlich und hat Zugriff auf alle im System gespeicherten Daten. Des Weiteren muss er uneingeschränkten Zugriff auf sämtliche im System vorhandenen Rechner haben, da er diese warten muss.

sonstige

Gerade im Bereich der Dienstverwaltung wären viele weitere Rollen denkbar und sinnvoll. Die Verwaltung der vom Ferienclub angebotenen Veranstaltungen könnte zum Beispiel durch ein eigenes Content Management System erfolgen, welches im Allgemeinen wieder aus mehreren Rollen besteht.

3.2 Anforderungen aus dem Beispielszenario

Da in dieser Arbeit der Gast mit dem mobilen Endgerät im Mittelpunkt steht, werden dessen Anwendungsfälle nun näher betrachtet. Dazu wurden Anwendungsfalldiagramme nach der UML (Unified Modelling Language)[1] Spezifikation erstellt.

3.2.1 Der Gast

Einchecken

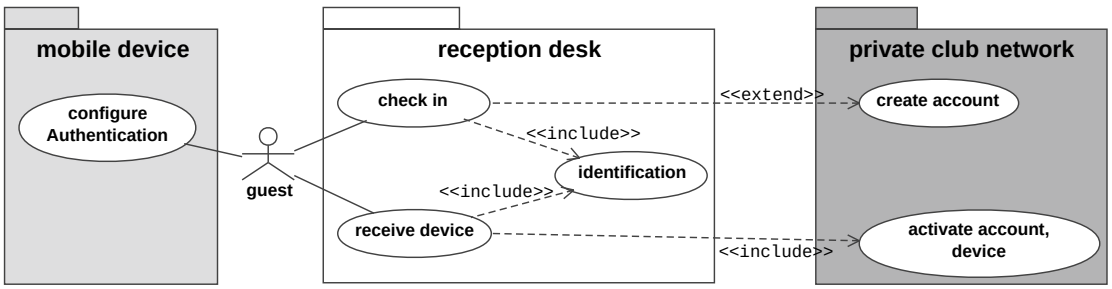

Abbildung 3.2: Anwendungsfalldiagramm: Gast - Einchecken

Die erste Interaktion des Urlaubers mit dem Club bildet das Einchecken (siehe Abbildung 3.2). Dabei muss er, wie im Beispielszenario beschrieben, an der Rezeption seine Identität nachweisen.

[1] Die Unified Modelling Language ist eine Spezifikation der OMG (Object Management Group) zur Visualisierung, Konstruktion und Dokumentation von Modellen für Softwaresysteme, Geschäftsmodelle und andere Nicht-Softwaresysteme (Dumke).

Nachdem er daraufhin den PDA übergeben bekommen hat, müssen die zur Authentifizierung benötigen Daten konfiguriert werden. Dies kann je nach Authentifizierungsverfahren anders aussehen.

Sollte der Urlauber nicht im voraus gebucht haben, so kann er trotzdem einen PDA zur Verfügung gestellt bekommen. Er muss allerdings in Kauf nehmen, dass er das Gerät nicht sofort ausgehändigt bekommt, sondern dass es erst vom PDA-Betreuer vorbereitet werden muss. Außerdem muss der Account des Gastes aktiviert werden. Dies bedeutet, dass im System vermerkt wird, dass der Gast überhaupt berechtigt ist, Dienste zu nutzen. Das Löschen dieses Vermerks wird in dieser Arbeit als deaktivieren bezeichnet.

Dienste nutzen

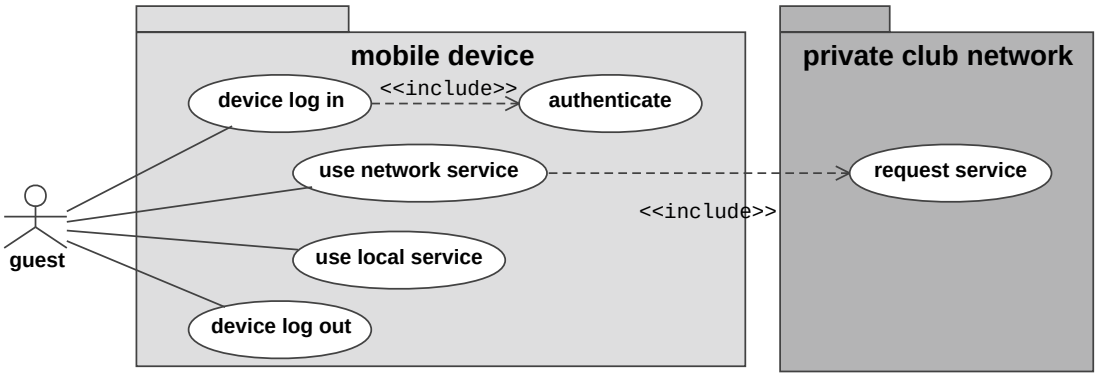

Abbildung 3.3: Anwendungsfalldiagramm: Gast - Dienste nutzen

Um einen Dienst nutzen zu können, muss sich der Benutzer an dem mobilen Gerät authentifizieren (siehe Abbildung 3.3). Diese Authentifizierung erfolgt lokal am Gerät, da der Urlauber in der Lage sein muss, den PDA auch außerhalb der Reichweite des Funknetzwerkes des Ferienclubs zu nutzen. Befindet sich der Urlauber außerhalb des erwähnten Netzes, so hat er nur Zugriff auf die Dienste, die sich lokal auf dem mobilen Endgerät befinden. Im System wird deshalb zwischen lokalen und Netzwerkdiensten unterschieden.

Sollte es sich bei der angeforderten Leistung um einen kritischen Dienst handeln, so muss diese Anfrage außerdem für die später stattfindende Abrechnung protokolliert werden.

Des Weiteren hat ein Benutzer die Möglichkeit, sich nach der Nutzung von Diensten explizit auszuloggen. Dadurch teilt er dem System mit, dass er vorerst keine weiteren Dienste nutzen möchte. Eine Nutzung des PDAs ist daraufhin erst nach einer erneuten Authentifizierung möglich.

Bei der Rezeption melden

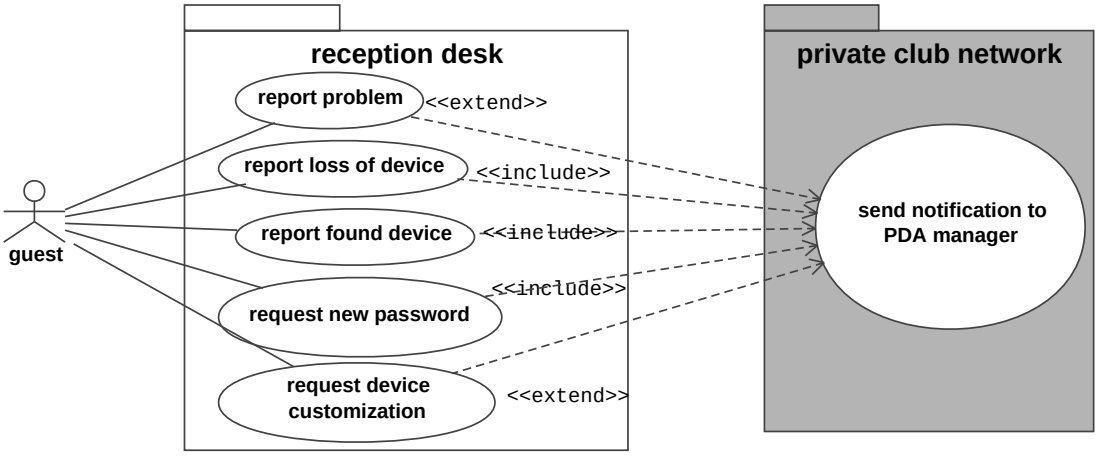

Abbildung 3.4: Anwendungsfalldiagramm: Gast - Unterstützung anfordern

Sollte der Benutzer irgendein Anliegen bezüglich seines mobilen Gerätes haben, so kann er sich damit an die Rezeption wenden (siehe Abbildung 3.4).

Diese Anliegen lassen sich grob in vier verschiedene Themenbereiche gliedern:

1. Dem Gast kommt das Gerät abhanden oder ein verloren gegangenes findet sich wieder an.
2. Der Benutzer hat Probleme oder Schwierigkeiten mit dem Gerät an sich. Dies kann der Fall sein, wenn er mit der bereitgestellten Software oder Hardware nicht zurechtkommt oder ein Soft- bzw. Hardwarefehler vorliegt
3. Der Benutzer vergißt sein Passwort.
4. Der Urlauber hat Änderungswünsche, die eine Anpassung des mobilen Gerätes benötigen.

Im zweiten, dritten und vierten Fall ist es unter Umständen nötig, dass der Gast den PDA an der Rezeption abgeben muss, damit diese das Gerät zur Problembehebung an den PDA-Betreuer weiterleiten kann. Kann dem Gast direkt an der Rezeption geholfen werden, so ist dies überflüssig.

Auschecken

Der Anwendungsfall des Auscheckens ist einfach gegliedert (siehe Abbildung 3.5). Der Urlauber meldet sich am Tag seiner Abreise an der Rezeption und gibt dort den PDA ab.

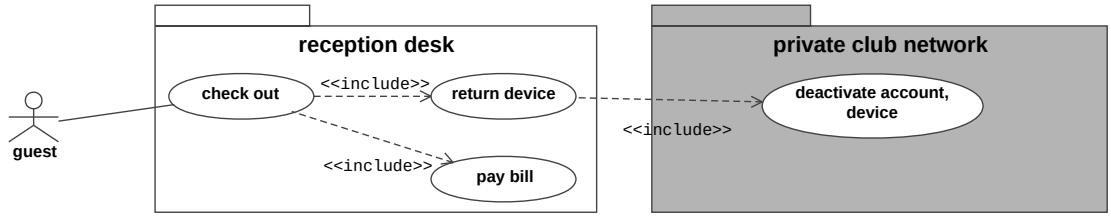

Abbildung 3.5: Anwendungsfalldiagramm: Gast - Auschecken

Des Weiteren muss er die angefallenen Kosten begleichen. Aus Sicherheitsgründen wird zusätzlich der Account des Gastes deaktiviert.

3.3 Anforderungen aus der Informatik

Um eine dauerhafte Architektur zu erstellen, die auch über einen längeren Zeitraum ihre Gültigkeit hat, sind neben den Anforderungen aus den Anwendungsfalldiagrammen noch generelle Anforderungen zu berücksichtigen, die sich in der Informatik über die Jahre herauskristallisiert haben. Es sind Anforderungen, die sich auf alle Bereiche der Informatik erstrecken und deren Berücksichtigung sich als überaus sinnvoll erwiesen hat. Dabei sind die einzelnen Anforderungen nicht als eigenständig zu betrachten, da sie sich gegenseitig beeinflussen können. So erhöht eine wohl überlegte Modularisierung eines Systems z.B. automatisch die Wartbarkeit.

3.3.1 Modularität

Unter Modularität versteht man die Unterteilung eines großen Themenkomplexes in mehrere kleinere eigenständige Teile (Module). Eine solche Aufteilung ist sinnvoll, da dadurch ein unproblematischer Austausch dieser Module möglich ist, ohne dass andere verändert oder angepasst werden müssen. Außerdem wird die Wartung des Gesamtsystems vereinfacht. Durch die Eigenständigkeit der Teilkomponenten wird zudem die Wiederverwendbarkeit gefördert.

3.3.2 Wiederverwendbarkeit

Das Einsetzten von Teilkomponenten eines Systems in einem anderen Produkt wird als Wiederverwendbarkeit bezeichnet. Wiederverwendbarkeit sollte angestrebt werden, da sich

dadurch die Entwicklungszeit neuer Produkte erheblich verkürzen lässt. Neben der Entwicklungszeit kann zusätzlich noch Zeit beim Testen der wiederverwendeten Teilkomponenten gespart werden, da diese bereits in dem Ursprungsprodukt getestet wurden.

3.3.3 Wartbarkeit und Erweiterbarkeit

Nach der Fertigstellung eines Systems bleibt es nicht aus, dass dieses mit der Zeit an neue Anforderungen angepasst werden muss. Es müssen Fehler behoben, neue Funktionen hinzugefügt und alte, nicht mehr erwünschte Eigenschaften entfernt werden. Andy Hunt und Dave Thomas haben diesen Umstand sehr treffend in ihrem Buch (Thomas und Hunt, 1999) beschrieben, in dem sie behaupten, dass Programmieren mehr mit Gartenarbeit als mit eigentlichem Ingenieurwesen zu vergleichen ist[2]. Durch diesen Umstand ist es oftmals notwendig, dass schnell und unkompliziert auf neue Anforderungen reagiert werden kann. Das Hinzufügen und Entfernen von Funktionalitäten, sowie das Beheben von Fehlern muss auch ohne Expertenwissen jederzeit möglich sein.

3.3.4 Portabilität

Unter Portabilität versteht man die Fähigkeit, ein System bzw. eine Architektur auf andere Umgebungen übertragen zu können. Im Bereich des Software Engineerings bezieht sich dies meistens auf die Übertragbarkeit von Software auf unterschiedliche Hardware Plattformen oder Betriebssysteme. Dadurch wird eine Unabhängigkeit gegenüber den unterschiedlichen Hard- und Software-Herstellern erreicht. Ein portables System kann demnach einfacher und schneller an Änderungen in der Firmenpolitik angepasst werden.

[2] „Rather than construction, programming is more like gardening.“

4 Sicherheitsanalysen

Ziel dieses Kapitels ist es, festzustellen, welche Daten im System gespeichert werden, wie wichtig diese sind und welchen Bedrohungen sie ausgesetzt sind. Die folgende Sicherheitsanalyse orientiert sich dabei weitestgehend an dem im IT-Grundschutzhandbuch des BSI (Bundesamt für Sicherheit in der Informationstechnik) vorgestellten Verfahren (BSI, 2002).

Im Mittelpunkt dieser Sicherheitsanalyse soll sowohl der PDA, als auch der Gast mit seinen dazugehörigen Daten stehen. Die Daten anderer Rollen werden hier nicht näher betrachtet, da sie nur im clubinternen Netz Verwendung finden und bereits ausreichend Literatur und Informationsmaterial zur Sicherung solcher Daten und Anwendungen existiert.

4.1 IT-Strukturanalyse

Die Aufgabe der IT-Strukturanalyse ist die Vorerhebung von Informationen, die für die Sicherheitsanalyse wichtig sind. Die IT-Strukturanalyse besteht nach dem BSI aus folgenden Teilaufgaben:

1. Netzplanerhebung
2. Komplexitätsreduktion durch Gruppenbildung
3. Erhebung der IT-Systeme
4. Erfassung der IT-Anwendungen und der zugehörigen Informationen

Abbildung 4.1 zeigt grob den Teil des Netzplans, der für den Gast interessant ist. Demnach besteht der betrachtete Bereich aus 4 Komponenten: dem PDA, dem Kommunikationskanal, einem AP (Access Point) und einem Dienstserver. Alle Komponenten können im betrachteten Szenario mehrfach vorhanden sein.

Eine Komplexitätsreduktion ist aufgrund der Einfachheit des Netzplans nicht notwendig. Auch die Erhebung der IT-Systeme kann entfallen, da es sich um ein einziges System handelt.

Als IT-Anwendungen werden hier die Anwendungen verstanden, mit denen der Gast seine Anwendungsfälle ausführen kann. Die dazugehörigen Daten ergeben sich aus den Informationen, welche für die einzelnen Anwendungsfälle benötigt werden (siehe Kapitel 3).

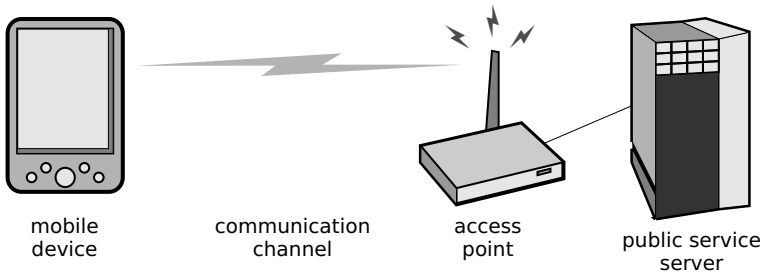

Abbildung 4.1: Systemkomponenten

4.1.1 Daten und Anwendungen des Gastes

In diesem Abschnitt soll eine Übersicht über die Daten erbracht werden, welche im System im Zusammenhang mit dem Gast gespeichert bzw. gesendet werden. Dies ist notwendig, um zu ermitteln, wie schützenswert diese im Einzelnen sind. Daraufhin können geeignete Sicherheitsvorkehrungen getroffen werden.

Die Art der Daten und deren Inhalt ergibt sich durch eine Betrachtung der einzelnen Anwendungsfälle aus dem Kapitel 3. Da es an dieser Stelle um IT-Anwendungen und ihre Daten gehen soll, werden nur die Aktionen aus der Anwendungsanalyse betrachtet, bei denen Verbindungslinien von Akteuren in Anwendungsfälle führen, die entweder im Paket mobile device oder im Paket public club network liegen. Anwendungsfälle, die sich im Paket reception desk befinden, werden nicht näher betrachtet. Das Ergebnis der Untersuchung zeigt Tabelle 4.1.

Aktion	Daten
use network service	Abrechnungsdaten, Zustandsdaten, PDA-Anwendungsdaten
use local service	PDA-Anwendungsdaten
device login	PDA-Authentifizierungsdaten, PDA-Anwendungsdaten
device logout	keine

Tabelle 4.1: Analyse: Anwendungsfälle und ihre Datenarten

Tabelle 4.2 zeigt eine Beschreibung, sowie den Inhalt der gefundenen Daten.

Zur lokalen Anmeldung des Benutzers am PDA werden zusätzlich noch die Daten aus Tabelle 4.3 benötigt.

4.2 Schutzbedarfsfeststellung

Ziel der Schutzbedarfsfeststellung ist es,

Datenart	Inhalt	Beschreibung
PDA-Anwendungsdaten	Kalender, Adressbuch, ...	Alle Daten, die auf dem PDA gespeichert werden. Dies sind vor allem Anwendungen mit den dazugehörigen Daten.
Abrechnungsdaten	Leistungen, Preise, Zeitstempel	Diese Daten umfassen alle vom Urlauber in Anspruch genommenen Dienste zusammen mit den dadurch verursachten Kosten. Die Daten werden für die spätere Abrechnung mit dem Kunden benötigt.
Zustandsdaten	aktiviert/deaktiviert	Daten, die Auskunft über den Status des jeweiligen Gastes machen.

Tabelle 4.2: Analyse: Beschreibung der Datenarten

Datenart	Inhalt	Beschreibung
PDA-Authentifizierungsdaten	Benutzername, Passwort, biometrische Daten, etc.	Diese Daten werden zur lokalen Authentifizierung des Urlaubers gegenüber dem PDA herangezogen.

Tabelle 4.3: Analyse: Authentifizierungsdaten

> "... für jede erfasste IT-Anwendung einschließlich ihrer Daten zu entscheiden, welchen Schutzbedarf sie bezüglich Vertraulichkeit, Integrität und Verfügbarkeit besitzt." (BSI, 2002, S. 41).

Dazu sind Auswirkungen zu untersuchen, die eine Beeinträchtigung der einzelnen IT-Anwendungen bzw. ihrer Daten mit sich bringen würden. Diese Vorgehensweise wird in der Literatur oft auch als Informationswertanalyse bezeichnet.

Die Einordnung der relevanten Daten erfolgt in so genannte Schutzbedarfskategorien. Sie geben an, wie groß die Schutzbedürfnisse der einzelnen Daten sind. Die hier verwendeten Schutzbedarfskategorien sind dem (BSI, 2002, S. 41) entnommen (siehe Tabelle 4.4).

Kategorie	Beschreibung
niedrig bis mittel	Die Schadensauswirkungen sind begrenzt und überschaubar.
hoch	Die Schadensauswirkungen können beträchtlich sein.
sehr hoch	Die Schadensauswirkungen können ein existenziell bedrohliches, katastrophales Ausmaß erreichen.

Tabelle 4.4: Schutzbedarfskategorien (BSI, 2002, S.41)

Um die Daten einzelnen Kategorien zuzuordnen, werden diese getrennt voneinander untersucht. Dabei wird ermittelt, welche Auswirkungen die Verletzung ihrer Vertraulichkeit, Verfügbarkeit und Integrität haben können.

4.2.1 Abrechnungsdaten

Ein Angreifer, der die Vertraulichkeit der Abrechnungsdaten verletzt, kann sich über alle vom Gast in Anspruch genommenen Dienste informieren. Zusätzlich erhält er Informationen darüber, wann diese Dienste genutzt wurden und welche Kosten sie verursacht haben. Im Ferienclub ist dies nicht weiter kritisch. Es sind jedoch Szenarios denkbar, in denen diese Informationen bereits wertvoll sein können.

Stehen die Abrechnungsdaten nicht zur Verfügung, so kann der Gast keine kritischen Dienste nutzen. Des Weiteren kann keine Abrechnung beim Auschecken des Urlaubers vorgenommen werden. Dies kann je nach Höhe der vom Gast verursachten Kosten wenig bis sehr kritisch sein. Je länger die Daten nicht zur Verfügung stehen, desto problematischer wird es für den Dienstanbieter. Sollte die Verfügbarkeit nur vorübergehend nicht gewährleistet werden können, besteht im gewählten Szenario jedoch die Möglichkeit, dem Gast später eine Rechnung zukommen zu lassen.

Die Verletzung der Integrität der Abrechnungsdaten kann schwerwiegende Auswirkungen haben, da bestehende Rechtsverbindlichkeiten gelöscht werden können. Es können also

kritische Dienste in Anspruch genommen werden, ohne dass einer der beiden vertragschließenden Parteien die vereinbarten Leistungen erbringt. Ein Gast könnte z.B. einen kostenpflichtigen Dienst nutzen, den er jedoch durch die Modifikation der Abrechnungsdaten nicht bezahlen muss. Es wäre außerdem denkbar, dass jemand neue Abrechnungsdaten einfügt und damit fiktive Rechtsverbindlichkeiten schafft. Dadurch könnten dem Gast Kosten für Dienste in Rechnung gestellt werden, die dieser nie in Anspruch genommen hat. Abschließend besteht die Möglichkeit, die vorhandenen Daten einfach nur zu verändern, um z.B. die entstandenen Kosten für einen Dienst zu erhöhen oder zu vermindern. Je nach angebotenen Diensten kann eine Verletzung der Integrität sehr hohe Schäden verursachen. Unter Umständen kann der finanzielle Schaden für die Institution existenzbedrohend sein, da die Abrechnung mit dem Kunden nach dem Beispielszenario erst am Ende seines Aufenthaltes geschieht.

4.2.2 PDA-Anwendungsdaten

Wie groß die Auswirkungen von Angriffen gegen die Vertraulichkeit der Anwendungsdaten ist, hängt im großen Maße davon ab, welche Informationen diese beinhalten. Geht es um die Anwendungen selber, so erhält der Angreifer Einsicht in die internen Abläufe und die Algorithmen des Programms. Verwendete Sicherheitsalgorithmen lägen damit offen. Betrachtet man nicht die Anwendungen, sondern deren Daten, so können diese sehr unterschiedliche Bedeutungen haben. Die Daten der Adressverwaltung und evtl. gespeicherte Abrechnungsdaten sind für einen Angreifer höchstwahrscheinlich interessanter, als die Daten eines installierten Spiels.

Sollte eine Anwendung oder ihre Daten nicht zur Verfügung stehen, so kann sie nicht genutzt werden. Dies bedeutet, dass der Gast auf dessen Nutzen verzichten muss. Da jedoch im Szenario alle Dienste auch ohne das mobile Endgerät abgewickelt werden können, sind die Auswirkungen eines solchen Angriffs recht gering.

Eine Veränderung der Anwendungsdaten kann schwere Folgen haben, da es damit möglich ist, die Algorithmen der Anwendungen zu manipulieren. Es wäre somit denkbar, dass ein Programm dahingehend verändert wird, dass es wertvolle Informationen, wie z.B. eingegebene Passwörter an Dritte weiterleitet. Des Weiteren besteht die Gefahr, dass die Anwendungen von Viren oder Trojanern befallen werden. Sollte es außerdem möglich sein, dass neue Anwendungen auf das mobile Endgerät aufgespielt werden, so können zusätzliche Gefahren entstehen. Ein solcher Angriff kann demnach beträchtliche Auswirkungen haben.

4.2.3 Zustandsdaten

Erhält jemand unberechtigte Einsicht in die Zustandsdaten, so kann dieser erfahren, ob ein Gast aktiviert ist oder nicht. Für Angreifer ist dieses Wissen sinnvoll, um z.B. festzustellen, ob ein PDA bereits gesperrt wurde oder nicht und damit, ob es sich überhaupt lohnt, einen Angriff auf den Account dieses Gastes durchzuführen oder nicht. Schaden entsteht durch dieses Wissen jedoch noch nicht.

Die Verletzung der Verfügbarkeit der Zustandsdaten hätte zur Folge, dass je nach Implementierung entweder Dienste genutzt werden können, obwohl der Gast gesperrt wurde, oder aber der Gast keine Dienste mehr in Anspruch nehmen kann, obwohl er eigentlich aktiviert ist.

Sollte die Integrität verletzt werden können, so kann ein Gast grundlos deaktiviert werden. Dies ist jedoch nicht weiter schlimm, da er sich jederzeit wieder an der Rezeption aktivieren lassen kann. Kritischer ist es, falls ein deaktivierter Gast widerrechtlich aktiviert wird. Dadurch könnte nicht mehr sichergestellt werden, dass ein Gast effektiv aus dem Clubnetzwerk ausgeschlossen wird.

4.2.4 PDA-Authentifizierungsdaten

Wenn die PDA-Authentifizierungsdaten von anderen Personen mitgelesen werden, könnten sich diese Zugang zu den auf dem PDA gespeicherten Daten verschaffen. Sollte es keine weiteren Sicherheitsmaßnahmen hinsichtlich der Nutzung von Diensten geben, so ist ein Angreifer in der Lage, sämtliche Dienste unter der Identität des Opfers zu nutzen. Dazu zählen vor allem kritische Dienste, welche eine rechtliche Bindung nach sich ziehen.

Sollte die Verfügbarkeit der PDA-Authentifizierungsdaten nicht gewährleistet sein, so kann sich der Gast nicht mehr an seinem PDA anmelden. Er kann somit weder auf die Daten des Gerätes zugreifen, noch vom Hotel angebotene Dienste nutzen. Es bleibt ihm nichts anderes übrig, als das Gerät zurück zur Rezeption zu bringen und um Hilfe zu bitten. Da alle Dienste auch ohne das mobile Endgerät abgewickelt werden können, sind auch in diesem Fall die Auswirkungen eines solchen Angriffs als gering einzustufen.

Eine Verletzung der Integrität der PDA-Authentifizierungsdaten hätte die gleichen Auswirkungen wie die in den beiden vorherigen Abschnitten geschilderten. Der Gast könnte sich nicht mehr am Gerät anmelden und der Angreifer könnte sich vollen Zugriff zu den auf dem PDA gespeicherten Daten verschaffen. Des Weiteren könnte der Angreifer sämtliche vom Ferienclub angebotenen Dienste unter der Identität des Gastes nutzen.

4.2.5 Zusammenfassung

Aus Gründen der Übersicht ist in Tabelle 4.5 noch einmal die Zuordnung aller Daten zu ihren jeweiligen Schutzbedarfskategorien zu sehen.

Datenart	Grundwert	Schutzbedarf
Abrechnungsdaten	Vertraulichkeit	mittel
	Verfügbarkeit	hoch
	Integrität	sehr hoch
Anwendungsdaten	Vertraulichkeit	mittel
	Verfügbarkeit	niedrig
	Integrität	hoch
Zustandsdaten	Vertraulichkeit	niedrig
	Verfügbarkeit	mittel
	Integrität	hoch
PDA-Authentifizierungsdaten	Vertraulichkeit	hoch
	Verfügbarkeit	niedrig
	Integrität	hoch

Tabelle 4.5: Ergebnis: Schutzbedarfsfeststellung

4.3 Bedrohungsanalyse

Nachdem die einzelnen Daten und Informationen des Systems betrachtet wurden, wird in diesem Kapitel genauer darauf eingegangen, welchen Bedrohungen die einzelnen Komponenten des Systems ausgesetzt sind.

Dieses Kapitel weicht von der Modellierung nach (BSI, 2002) ab, da mobile Endgeräte darin bisher nur unzureichend betrachtet wurden. Auch sind die Bausteine des BSI für die Modellierung teilweise zu speziell und nicht ohne weiteres auf das zu erstellende System übertragbar. Stattdessen wird eine getrennte Bedrohungsanalyse und anschließend der Entwurf einer Sicherheitsarchitektur durchgeführt. Die Bedrohungsanalyse erfolgt mit Hilfe von Bedrohungsbäumen, wie sie in (Schneier, 1999) definiert sind. Der Wurzelknoten definiert dabei ein mögliches Angriffsziel, welches durch die Unterknoten erreicht werden kann. Diese Unterknoten bilden dann wiederum eigene Ziele für weitere Angriffe. Die Unterknoten können entweder UND- oder ODER-verknüpft sein, je nachdem, ob zum Erreichen des Ziels alle oder mindestens ein Unterknoten erfüllt sein müssen. In den hier erarbeiteten Bedrohungsbäumen sind alle nicht näher bezeichneten Knoten als ODER-Knoten zu betrachten.

Bei der Analyse werden alle Komponenten des in Abbildung 4.1 abgebildeten Netzplans untersucht. Hauptaugenmerk liegt hier jedoch auf dem PDA und dem drahtlosen Netz. Ac-

cess Point und Server wurden zwar ebenfalls untersucht, werden jedoch nur kurz skizziert, da eine genau Betrachtung den Rahmen dieser Arbeit sprengen würde.

Des Weiteren werden hauptsächlich technische Bedrohungen analysiert. Bedrohungen, wie sie zum Beispiel durch höhere Gewalt oder organisatorische Mängel entstehen können, werden nicht näher betrachtet. Um die Komplexität der einzelnen Bedrohungsbäumen zu verringern, werden nur die Bedrohungen aufgeführt, die in diesem Szenario am wahrscheinlichsten sind. Details zu den einzelnen Knoten finden sich in den entsprechenden Textabschnitten. Die Nummerierung der Abschnitte orientiert sich dabei an den Knoten der einzelnen Bedrohungsbäume.

4.3.1 PDA

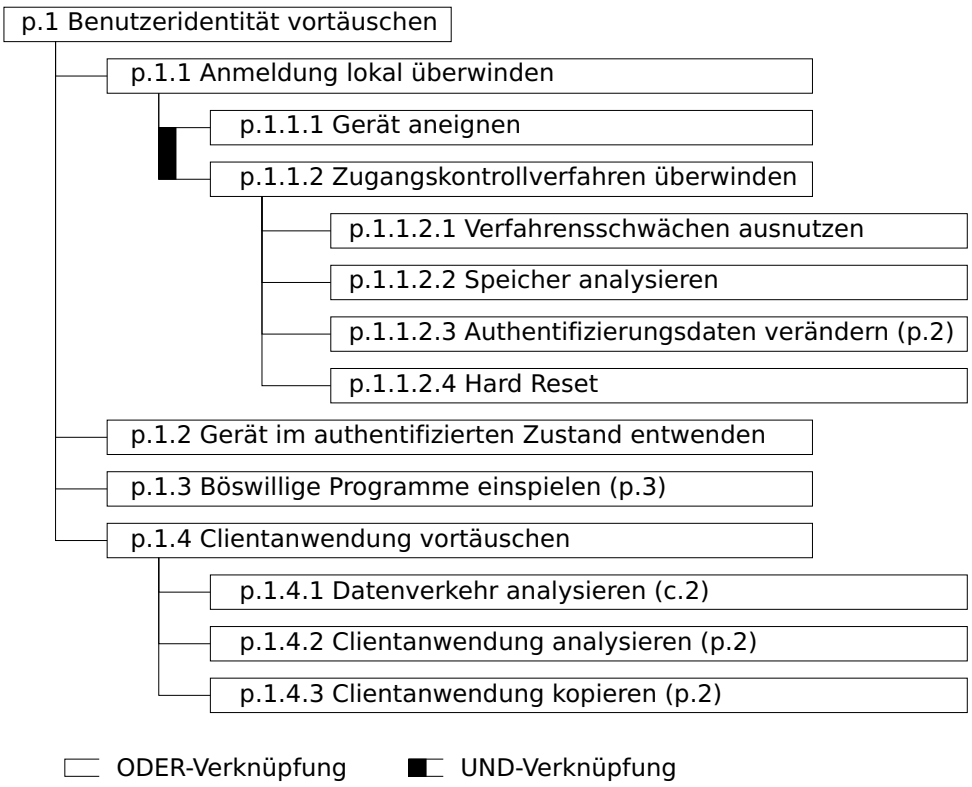

Abbildung 4.2: Bedrohungsbaum: p.1 Benutzeridentität vortäuschen

p.1 Benutzeridentität

Die nachfolgenden Angriffe zielen alle darauf ab, die Identität des PDA-Benutzers vorzutäuschen. Es sollen Aktionen ausgeführt werden, welche das System als zu einem bestimmten

Benutzer zugehörig einstuft. Abbildung 4.2 zeigt den dazugehörigen Bedrohungsbaum.

p.1.1 Anmeldung überwinden

Durch das Überwinden des lokal auf dem PDA laufenden Anmeldeverfahrens ist es möglich, auf dem Gerät alle die Funktionen auszuführen, die der eigentliche Benutzer ebenfalls ausführen kann. Ebenso kann der Angreifer alle Daten einsehen, die der Benutzer auf dem PDA gespeichert hat. Um die Anmeldung zu überwinden, muss sich der Angreifer sowohl den PDA aneignen, als auch das lokale Zugangskontrollverfahren umgehen.

p.1.1.1 Gerät aneignen

Ein solcher Angriff lässt sich gerade bei PDAs aufgrund ihrer Bauart einfach ausführen (Ghosh und Swaminatha, 2001). Einer Studie nach sollen allein im Jahr 2002 250.000 PDAs und Handhelds verloren gegangen sein (David, 2002). Eine anderen Quelle besagt, dass die Verlustrate von PDAs bei ca. 30% liegt (Lyon, 2002). Die Aneignung des Gerätes kann der Angreifer auf verschiedene Arten realisieren. Nach (Eckert, 2001b) kann dies durch Diebstahl, unbeaufsichtigtes Stehenlassen, Verlust oder Weitergabe geschehen. Solche Angriffe sind besonders im gewählten Szenario leicht vorstellbar, da die meisten Gäste wohl weder das notwendige Sicherheitsbewußtsein, noch die Erfahrung im Umgang mit einem solchen Gerät haben.

p.1.1.2 Zugangskontrollverfahren überwinden

Das Zugangskontrollverfahren muss überwunden werden, damit der Angreifer den PDA nutzen kann. Dazu bieten sich ihm vier verschiedene Möglichkeiten, die nun im Einzelnen diskutiert werden sollen.

p.1.1.2.1 Verfahrensschwächen ausnutzen

Bevor auf die Schwächen der einzelnen Zugangskontrollverfahren eingegangen wird, soll kurz erläutert werden, was der Begriff bedeutet. Zugangskontrollverfahren beschäftigen sich mit der Identifikation und Authentifizierung der Benutzer eines Systems. Während der Benutzer bei der Identifikation dem System mitteilt, wer er ist, befasst sich die Authentifizierung mit Methoden, mit denen ein Benutzer beweisen kann, dass er derjenige ist, für den er sich ausgibt (Russel und Gangemi, 1991).

Die Authentifizierung kann durch drei verschiedene Arten erfolgen.

1. Durch etwas, das man weiß,

2. etwas, das man hat oder
3. etwas, das man ist.

Zur ersteren gehören alle diejenigen Verfahren, bei denen der Benutzer seine Identität dadurch beweisen muss, dass er ein geheimes Wissen hat. Ein typisches Beispiel für ein solches Verfahren ist die Passwortabfrage. Es wird dabei davon ausgegangen, dass eine Person ein Passwort bekommt, um sich dem System gegenüber zu authentifizieren. Niemand sonst außer dieser Person darf das Passwort kennen. Probleme dieser Methoden sind, dass die verwendeten Daten weitergegeben werden können. Werden sie aufgeschrieben, können sie von Dritten gelesen oder gestohlen werden. Ist das Wissen zu einfach gewählt, lässt es sich erraten oder durch Ausprobieren bzw. systematisches Vorgehen herausfinden. Oftmals lassen sich die benötigten Daten auch durch die Beobachtung des Gastes bei der Eingabe ausspionieren. Begünstigt werden solche Angriffe oft durch ein fehlendes Sicherheitsbewusstsein des Benutzers (Karygiannis und Owens, 2002, 5-26). Dieser Faktor ist gerade in Bezug auf das hier vorliegende Szenario nicht zu unterschätzen.

Beim zweiten Verfahren erhält jeder Benutzer einen Gegenstand, den er zur Authentifizierung am System benötigt. Dies kann dadurch geschehen, dass dieser Gegenstand dem System direkt präsentiert wird (Ausweis, Smart Cards, ...), oder dass dem System das Vorhandensein dieses Gegenstandes durch Anwendung bewiesen wird (Schlüssel, Tokens, ...). Auch diese Authentifizierungsmethode leidet unter dem Problem, dass die benötigten Gegenstände geklaut werden können. Des Weiteren können viele Gegenstände dupliziert werden.

Die dritte Methode basiert auf physiologischen und verhaltensbasierten Merkmalen der einzelnen Benutzer. Darunter fallen zum Beispiel Finger- und Handabdrücke, sowie Netzhautmuster, Sprach- und Unterschriftsmerkmale aber auch so genannte Keystrokes, die das Tippverhalten der einzelnen Benutzer charakterisieren. Die Verfahren dieser dritten Methode werden auch als biometrische Verfahren bezeichnet. Sie lassen sich nur schwer umgehen. Es muss versucht werden, das zur Prüfung verwendete biometrische Merkmal nachzuahmen, oder aber sich dieses biometrische Merkmal zu verschaffen. Die Nachahmung eines biometrischen Merkmals kann zum Erfolg führen, da eine so genannte FAR (False Acceptance Rate)[1] vielen biometrischen Verfahren eigen ist (Krause und Tipton, 1998). Ein weiteres Problem ist, dass solche biometrische Verfahren von Benutzern oft nicht akzeptiert werden. Dies liegt u.a. an der Tatsache, dass sich Menschen ungern vermessen lassen und dass bestimmte Verfahren sogar als gefährlich eingestuft werden (Russel und Gangemi, 1991). Eine zusätzliche Problematik wird dadurch aufgeworfen, dass sich biometrische Merkmale mit der Zeit verändern können. Dadurch müssen Toleranzschwellen ins System eingebaut werden, die zu Lasten der Sicherheit gehen.

In diesem Zusammenhang wird nach (Kersten, 1991) zwischen der reinen Funktionalität und der Qualität eines Verfahrens unterschieden. Unter der Funktionalität wird dabei das

[1] Unter False Acceptance Rate versteht man die Rate, mit der Betrüger als gültige Benutzer erkannt werden.

reine Vorhandensein eines Mechanismus bezeichnet. Die Qualität gibt dagegen die „Standfestigkeit“ und damit auch die Angreifbarkeit eines Verfahrens an.

p.1.1.2.2 Speicher analysieren

Da die Daten, anhand denen das Authentifizierungsverfahren prüft, ob einem Benutzer Zugang gewährt wird oder nicht, auf dem PDA gespeichert sein müssen, kann eine Analyse des Speichers wertvolle Informationen liefern. Die Analyse ist dabei je nach verwendetem Speicher unterschiedlich schwierig. Sollte das mobile Endgerät leicht auswechselbare, normierte Speichermodule für die Datenhaltung verwenden, wie z.B. Compact Flash[2] Karten, so kann ein solcher Angriff wesentlich erleichtert werden (siehe Abbildung 4.3).

Abbildung 4.3: Speichererweiterungsmodul für den IPAQ - (PocketPCCentral)

Es soll jedoch nicht verschwiegen werden, dass ein solcher Angriff hauptsächlich für wissensbasierte Zugangsverfahren geeignet ist. Biometrische Merkmale anhand einer Datenanalyse nachzubilden kann sehr aufwendig sein.

p.1.1.2.3 Authentifizierungsdaten verändern

Nicht nur die Analyse der eben erwähnten Daten kann zum Erfolg führen. Wesentlich einfacher ist das Austauschen oder Verändern der verwendeten Daten durch eigene, bekannte Daten. Dazu muss schreibend auf den Speicher zugegriffen werden. Der Vorteil einer solchen Methode ist, dass auf diese Weise auch biometrische Verfahren leicht überwunden werden können. Wie Daten oder Programme verändert werden können, wird in Abschnitt p.2 behandelt.

[2] http://www.compactflash.org/

p.1.1.2.4 Hard Reset

Zusätzlich zu den gerade genannten Angriffen, lässt sich bei vielen PDAs ein Hard Reset durchführen. Nach einem solchen Reset werden normalerweise alle Authentifizierungsdaten zurückgesetzt und ein Angreifer kann somit Zugriff auf das Gerät erhalten, ohne sich authentifizieren zu müssen. Das Bedrohungspotenzial durch einen Hard Reset hängt stark davon ab, welche zusätzlichen Daten und Anwendungen dadurch gelöscht und/oder verändert werden.

p.1.2. Gerät im authentifizierten Zustand entwenden

Gerade in sehr belebten Umgebungen, wie dies im gewählten Szenario der Fall ist, kann es vorkommen, dass ein unbeobachteter PDA in einem Zustand entwendet wird, in dem die lokale Authentifizierung durch den Gast bereits erfolgt ist. Da davon ausgegangen werden kann, dass viele Gäste im Umgang mit einem PDA ungeübt sind und dadurch anzunehmen ist, dass auch deren Sicherheitsbewusstsein gering ist, ist die Gefahr einer solchen Attacke relativ hoch.

p.1.3. Böswillige Programme einspielen

Durch das Einspielen von böswilligen Programmen kann ein Angreifer Aktionen unter der Identität des eigentlichen Benutzers ausführen. Der Vielfältigkeit der Aktionen sind dabei keine Grenzen gesetzt. Es kann sich um einfache Viren handeln, aber auch um Würmer oder trojanische Pferde. Angriffe können auch durch das Installieren von Fernwartungssoftware oder Key-Logging Software, die sämtliche Eingaben des Benutzers mitprotokollieren, realisiert werden. Für weitere Bedrohungen durch böswillige Programme sei hier auf das Buch (Thaller, 1993) verwiesen. Wie solche Programme eingespielt werden können, wird in einem gesonderten Bedrohungsbaum behandelt.

p.1.4. Clientanwendung vortäuschen

Die Identität eines Benutzers lässt sich außerdem vortäuschen, indem der Angreifer dem Server die Clientanwendung des Benutzers vortäuscht. Als Clientanwendung wird in dieser Arbeit die Software verstanden, die es dem Benutzer ermöglicht, Netzwerkdienste in Anspruch zu nehmen. Durch einen solchen Angriff lässt sich allerdings nur der Server über die Identität des Angreifers täuschen. Der PDA bleibt davon unberührt. Wie das Clientprogramm erfolgreich vorgetäuscht werden kann, wird in den folgenden Abschnitten näher untersucht.

p.1.4.1. Datenverkehr analysieren

Ein Weg, um die Funktionen des Clients eines Benutzers nachzuahmen, besteht darin, den Datenverkehr zwischen Client und Server zu untersuchen. Durch die Analyse der gesendeten Daten können unter Umständen Rückschlüsse auf das verwendete Protokoll und die Bedeutung der einzelnen gesendeten Nachrichten getätigt werden. Daraufhin kann ein Angreifer versuchen, den Client selbst nachzubauen. Das Analysieren der Funktionsweise einer Anwendung, bei der man keine Einsicht in den Sourcecode hat, wird auch als Software Reverse Engineering bezeichnet (Peikari und Chuvakin, 2004).

p.1.4.2. Clientanwendung analysieren

Nicht nur der Datenverkehr zwischen Client und Server kann Aufschluss über das Verhalten und die Funktionsweise eines Programmes liefern. Das Dekompilieren der Clientanwendung selbst kann ebenfalls aufschlussreich sein. Ist die Funktionsweise der Anwendung entschlüsselt, so ist der Angreifer in der Lage, diese nachzubauen. Dieser Nachbau kann es ermöglichen, Nachrichten an den Server zu schicken, ohne dass dieser einen Unterschied zwischen der Kommunikation mit dem echten Benutzer und dem Angreifer bemerkt.

Programme, die das Software Reverse Engineering unterstützen, werden in (Peikari und Chuvakin, 2004) vorgestellt. Wichtig ist jedoch, dass solche Angriffe nur durchführbar sind, wenn mindestens ein lesender Zugriff auf die Clientanwendung gegeben ist und geeigneten Tools angewendet werden können.

p.1.4.3. Clientanwendung kopieren

Eine recht einfache Methode besteht darin, die Clientanwendung zu kopieren. Dazu wird ebenfalls ein zumindest lesender Zugriff auf die Clientanwendung, sowie ein schreibender Zugriff auf ein Speichermedium, benötigt. Für den Angreifer bedeutet diese Methode, dass er einen Computer besitzen muss, der das kopierte Programm zur Ausführung bringen kann. Auch diese Angriffsart wird durch leicht entfernbare, externe Speichermedien begünstigt.

p.2 Programme oder Daten verändern

Das Verändern von Programmen ist gefährlich, da es dadurch möglich ist, deren Verhalten zu modifizieren, um nicht genehmigte Funktionen auszuführen. Behindert dieser neue Code die normale Arbeit des Programms nicht, so spricht man auch von einem Trojanischen Pferd (Thaller, 1993).

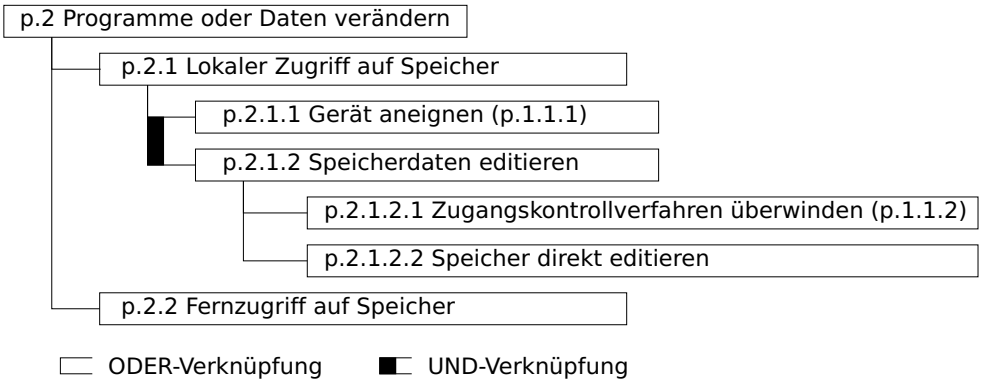

Abbildung 4.4: Bedrohungsbaum: p.2 Programme oder Daten verändern

Veränderungen an den auf dem PDA gespeicherten Daten können ebenfalls weitreichende Folgen haben. Je nachdem, welche Daten verändert oder eingesehen werden, kann eine solche Bedrohung mehr oder weniger gefährlich sein. Welche Auswirkungen dies im Einzelnen sind, wurde bereits in Kapitel 4.2 behandelt. Bisher nicht beachtet wurden Änderungen an Betriebssystemeinstellungen. Diese stellen eine besondere Gefahr dar, da durch sie andere Angriffe ermöglicht werden können. Beispiele hierfür sind das Öffnen gesperrter Kommunikationskanäle, sowie die Deaktivierung des lokalen Zugangskontrollverfahrens.

Der Zugriff auf die Daten des PDAs kann zum einen lokal, oder aus der Ferne geschehen. Wie diese beiden Arten von Angriffen realisiert werden können, wird in den folgenden Abschnitten beschrieben.

Der zu dieser Bedrohung gehörenden Bedrohungsbaum findet sich in Abbildung 4.4.

p.2.1 Lokaler Zugriff auf Speicher

Lokale Änderungen unterscheiden sich von den Angriffen aus der Ferne dadurch, dass das Gerät selbst vorhanden sein muss. Vorteil eines solchen Angriffs ist, dass er selbst dann durchgeführt werden kann, wenn sämtliche Kommunikationskanäle des Gerätes gesperrt sind.

p.2.1.1 Gerät aneignen

Auf welche verschiedenen Weisen ein Angreifer in den Besitz des mobilen Gerätes kommen kann, wurde bereits unter p.1.1.1 behandelt.

p.2.1.2 Speicherdaten editieren

Um die Speicherdaten selbst zu editieren, bieten sich dem Angreifer zwei verschiedene Methoden an.

p.2.1.2.1 Zugangskontrollverfahren überwinden

Zum einen kann das Zugangskontrollverfahren überwunden werden. Dies gibt dem Angreifer die Möglichkeit, auf dem mobilen Endgerät die gleichen Veränderungen durchzuführen, die der jeweilige Benutzer durchführen könnte. Wie erfolgversprechend so ein Angriff ist, hängt demnach sehr stark von den Rechten des Benutzers ab. Möglichkeiten zur Überwindung des Zugangskontrollverfahrens wurden bereits in p.1.1.2 behandelt.

p.2.1.2.2 Speicher direkt editieren

Zum anderen kann der Angreifer versuchen, den Speicher direkt zu editieren. Dabei spielt, wie bereits erwähnt, die Art des Speichers eine große Rolle. Auch hier sind Angriffe auf genormte Speicherbausteine, für die es frei verfügbare Lese- und Schreibgeräte gibt, wesentlich einfacher zu realisieren als auf Spezialanfertigungen.

p.2.2 Fernzugriff auf Speicher

Um aus der Ferne Daten oder Programme des Gerätes zu verändern oder auszuführen, muss der Angreifer in der Lage sein, einen der vorhandenen Kommunikationskanäle zu missbrauchen. Sollte auf dem PDA kein Dienst zur Verfügung stehen, welcher vom Angreifer für diesen Zweck missbraucht werden kann, so muss dieser erst einmal ein externes Programm auf das Gerät des Benutzers einspielen. Wie dies erreicht werden kann, zeigt der nächste Abschnitt.

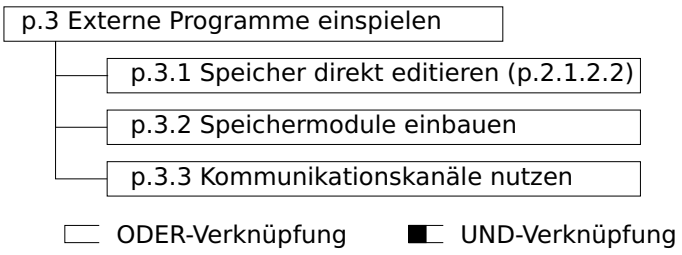

Abbildung 4.5: Bedrohungsbaum: p.3 Externe Programme einspielen

p.3 Externe Programme einspielen

Das Einspielen von Software auf den PDA stellt eine Bedrohung dar, da nicht hundertprozentig sichergestellt werden kann, dass diese Programme keinen Schaden anrichten. Dabei kann ein Schaden sowohl durch böswillige Programme wie z.B. Viren und Trojanische Pferde, als auch durch Fehlfunktionen entstehen. Unter Fehlfunktionen werden hier unter anderem das zum Absturz bringen des Gerätes an sich und das Einbringen von neuen Sicherheitslücken verstanden. Bisher sind noch nicht viele Viren oder andere böswillige Programme für PDAs bekannt, doch warnen Experten, dass sich dieser Zustand in Zukunft ändern wird (BBCvirus, 2003). Abbildung 4.5 stellt den Bedrohungsbaum dar, der zu diesem Angriff gehört.

p.3.1 Speicher direkt editieren

Wie bereits in p.2.1.2 geschildert, kann ein Angriff darauf zielen, den Speicher des Gerätes direkt zu editieren. Auf diese Weise ist es nicht nur möglich, bestehende Daten und Programme zu verändern, sondern auch neue Daten und Programme auf den PDA aufzuspielen. Um einen solchen Angriff durchführen zu können, muss sich der Angreifer vorher das Gerät aneignen (siehe p.1.1.1).

p.3.2 Speichermodule einbauen

Der Einbau neuer Speichermodule wird dadurch vereinfacht, dass viele Geräte so gebaut werden, dass ihr Speicher leicht erweitert werden kann. Dies äußert sich dadurch, dass die zur Erweiterung benutzten Speicherbausteine oftmals frei verfügbar und die Erweiterungsschächte leicht zugänglich sind. In letzter Zeit werden dabei vermehrt externe Speichererweiterungsmodule wie Compact Flash, etc. eingesetzt. Das macht diese Art von Angriffen besonders einfach. Besonders kritisch sind in diesem Zusammenhang Betriebssysteme, die neue Speichermedien automatisch erkennen und in das System einbinden, ohne dass dieses Verhalten verändert werden kann. Einige Hersteller gehen sogar soweit, dass Anwendungen auf Speicherkarten definiert werden können, die automatisch beim Einführen in das Gerät gestartet werden (Dedo, 2004).

p.3.3 Kommunikationskanäle nutzen

Da das mobile Endgerät mit dem Dienstserver des Ferienclubs kommunizieren muss, ist zumindest ein Kommunikationskanal zeitweise geöffnet. Es kann jedoch noch weitere Kanäle geben, über die eine Kommunikation stattfinden kann. Gerade in den letzten Jahren hat die Konnektivität der einzelnen mobilen Geräte stark zugenommen. So verfügen viele bereits

neben dem üblichen Synchronisationskabel und einer Infrarotschnittstelle auch über Technologien wie Bluetooth und WLAN. Alle diese Kommunikationskanäle können theoretisch dazu verwendet werden, externe Programme auf das Gerät einzuschleusen.

Besonders zu berücksichtigen ist hier wiederum das häufig fehlende Sicherheitsbewusstsein der Benutzer (Price, 2003). Wie unvorsichtig Benutzer mit dem Einspielen unbekannter Programme sind, haben Viren bzw. Würmer wie Mydoom (Kuri, 2004), Blaster (Bachfeld, 2003) und Loveletter (Donath, 2000) gezeigt.

4.3.2 Kommunikationskanal

An dieser Stelle wird nur aufgrund des gewählten Szenarios nur auf drahtlose Kommunikationskanäle eingegangen. Die Untergliederung des Baumes findet anhand der Kategorien Verfügbarkeit, Vertraulichkeit und Integrität statt. Welche Datenarten durch solche Angriffe betroffen sind, hängt sowohl von dem verwendeten Protokoll, als auch von der Verteilung der Daten zwischen Client und Server ab. Dadurch lassen sich keine genauen Angaben über die Gefährlichkeit der einzelnen Angriffe machen. Sollte es sich bei den Daten um die in Kapitel 4.1.1 vorgestellten handeln, so kann dafür Tabelle 4.5 zur Hilfe genommen werden.

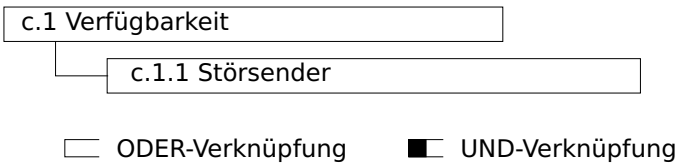

Abbildung 4.6: Bedrohungsbaum: c.1 Verfügbarkeit

c.1 Verfügbarkeit

Sollte die Verfügbarkeit des Kommunikationskanals angegriffen werden, so können keine Nachrichten mehr zwischen Client und dem Server ausgetauscht werden. Der Benutzer kann somit nur noch lokal installierte Programme bzw. Dienste nutzen. Der Bedrohungsbaum dazu ist relativ klein und in Abbildung 4.6 dargestellt.

c.1.1 Störsender

Die wohl einfachste technische Möglichkeit, die Verfügbarkeit eines drahtlosen Netzes anzugreifen, besteht in der Verwendung von Störsendern. Dazu wird bei Funkverbindungen

lediglich ein Gerät benötigt, welches auf der für die Übertragung genutzten Frequenz Störungen sendet. Besonders WLANs sind durch einen solchen Angriff bedroht. Dies liegt nicht nur an dem zugrunde liegenden Konzept, sondern auch an der Tatsache, dass sich die benötigten Komponenten zum Bau eines solchen Senders relativ leicht beschaffen lassen (Stahlberg, 2000).

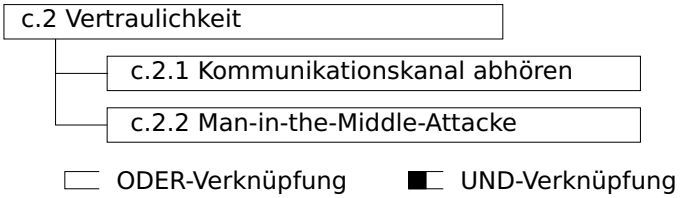

Abbildung 4.7: Bedrohungsbaum: c.2 Vertraulichkeit

c.2 Vertraulichkeit

Wie gefährlich Angriffe auf die Vertraulichkeit des Kommunikationskanals sind, hängt von den gesendeten Daten ab. Das Mitlesen der Abrechnungsdaten, sowie der PDA-Authentifizierungsdaten wäre nach der Analyse aus Kapitel 4.2 besonders gefährlich.

Der Bedrohungsbaum findet sich in Abbildung 4.7.

c.2.1 Kommunikationskanal abhören

Ein solcher Angriff ist in drahtlosen Netzen besonders einfach zu realisieren. Das liegt unter anderem daran, dass die Daten meistens nicht gerichtet, sondern vom Sender in alle Richtungen verbreitet werden. Daher muss der Angreifer sich nicht unbedingt in unmittelbarer Nähe der eigentlichen Kommunikationslinie befinden. Abbildung 4.8 zeigt anschaulich, welche Reichweite ein einzelner Access Point haben kann.

Ein weiterer Grund ist, dass die Preise für die benötigten Empfangsgeräte für drahtlose Technologien immer weiter fallen und sich somit die Kosten eines solchen Angriffs ebenfalls in Grenzen halten.

c.2.2 Man-in-the-Middle-Attacke

Einen Sonderfall des Abhörens eines Kommunikationssignals stellt die so genannte Man-in-the-Middle-Attacke dar. Dazu schaltet sich der Angreifer zwischen die Kommunikation von Client und Server, ohne dass diese dies bemerken. Dies lässt sich dadurch realisieren, dass

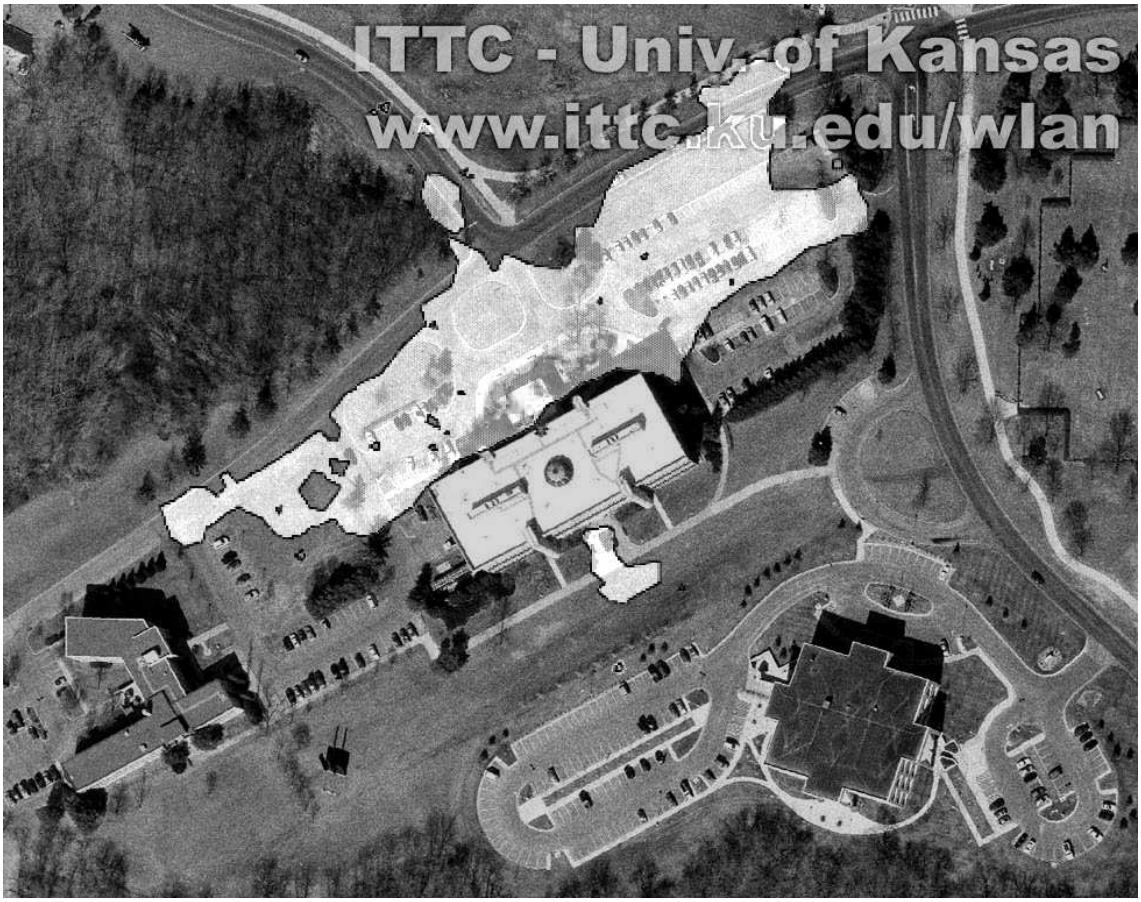

Abbildung 4.8: WLAN-Ausbreitung eines APs (WLANVisual, 2002)

der Angreifer sämtliche gesendeten Nachrichten abfängt und diese dann an den eigentlichen Empfänger weiterleitet. Da dies auch mit den Antwortnachrichten geschieht, können weder Sender noch Empfänger feststellen, ob die Nachrichten zwischendurch abgefangen wurden, oder nicht. Das Besondere an einem solchen Angriff ist, dass dadurch unter Umständen auch asynchron verschlüsselte Verbindungen bzw. Verbindungen, bei denen zur Authentifizierung ein asynchrones Verfahren angewendet wird, abgehört werden können. Grundvoraussetzung für einen solchen Angriff ist, dass weder der Client noch der Server eine verlässliche Möglichkeit hat, die Identität der Gegenstelle eindeutig zu identifizieren. Wie ein solcher Angriff genau aussehen kann, zeigt Abbildung 4.9.

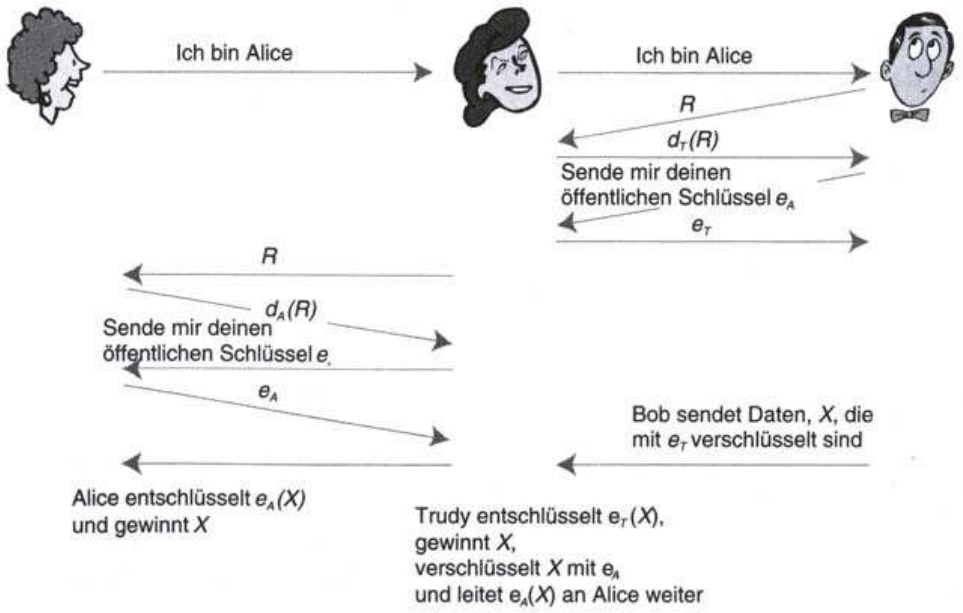

Abbildung 4.9: Man-in-the-Middle-Attacke (Kurose und Ross, 2001, S. 574)

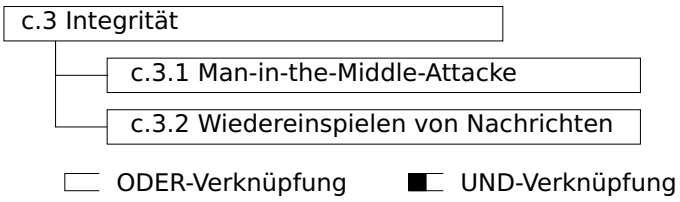

Abbildung 4.10: Bedrohungsbaum: c.3 Integrität

c.3 Integrität

Durch eine Verletzung der Integrität eines Kommunikationskanals können sämtliche ausgetauschte Nachrichten verändert werden. Dies ist insbesondere im Hinblick auf die Abrechnungsdaten sehr gefährlich. Wie solche Angriffe aussehen können, wird in den folgenden Abschnitten beschrieben. Abbildung 4.10 zeigt den relevanten Bedrohungsbaum.

c.3.1 Man-in-the-Middle-Attacke

Bei einer Man-in-the-Middle-Attacke hat der Angreifer neben dem rein lesenden Zugriff auch die Möglichkeit, die Nachricht zu verändern. Dadurch können z.B. Parameter beim Nutzen von Diensten verändert werden.

c.3.2 Wiedereinspielen von Nachrichten

Des Weiteren können Angreifer Nachrichten mitschneiden und zu einem späteren Zeitpunkt wieder einspielen. Dabei wird zwar die Nachricht als solche nicht verändert, aber das Ergebnis kann eine Änderung im Datenbestand des Servers, bzw. Clients hervorrufen. Es wäre zum Beispiel möglich, die Nutzung eines Dienstes durch einen Gast mitzulesen und diese Nachricht später erneut einzuspielen. Handelt es sich dabei um einen kritischen Dienst, so würde dem Gast unter Umständen der Dienst ein zweites Mal berechnet werden.

4.3.3 Access Point und Server

Sowohl auf den Access Point, als auch auf den Server soll hier nur kurz eingegangen werden, da bereits genügend Literatur über dieses Thema existiert (z.B. (Russell u. a., 2003)). Es sollen deshalb nur die Bedrohungen vorgestellt werden, die sich speziell auf die Eigenschaften des Dienstservers beziehen.

Serveranwendung vortäuschen

In Kapitel 4.3.1 wurde das Vortäuschen der Clientanwendung behandelt. Die gleichen Bedrohungen gehen auch von nachgebildete Serveranwendungen aus. Eine Datenanalyse der gesendeten Nachricht kann dabei als Grundlage für den Bau einer solchen Anwendung dienen. Aber auch das bereits erwähnte Analysieren bzw. Kopieren der Software ist denkbar.

4.3.4 sonstige Angriffe

Einige Bedrohungen wurden bisher noch nicht behandelt, da sie sich entweder auf alle Komponenten beziehen, oder sich nicht genau einzelnen Komponenten zuordnen ließen.

Bugs

Bedrohungen, denen fast jede Komponente ausgesetzt ist, sind Bugs [3]. Es gibt dutzende von Internetseiten, auf denen täglich neue Sicherheitslöcher verschiedener Softwareprodukte veröffentlicht werden. Wie die Sicherheit durch solche Bugs gefährdet werden kann, hängt vom eigentlichen Fehler ab. Denkbar sind Fehler, die die Komponente zum Absturz bringen, sowie Fehler, die es dem Angreifer ermöglichen, Administrationsrechte auf der jeweiligen Komponente zu erhalten. Deshalb darf diese Bedrohung auf keinen Fall unterschätzt werden.

Abstreitbarkeit

Eine weitere Bedrohung stellt die Abstreitbarkeit von eingegangenen Rechtsverbindlichkeiten dar. Es muss eine Lösung gefunden werden, die verhindert, dass eine der beiden vertragsschließenden Parteien bei der Nutzung eines kritischen Dienstes behaupten kann, dass sie den Vertrag niemals eingegangen ist. Dies trifft im gewählten Szenario insbesondere auf den Gast zu, da davon ausgegangen werden kann, dass der Ferienclub selbst vertrauenswürdig ist.

4.3.5 Zusammenfassung

Zusammenfassend lässt sich feststellen, dass das System einer ganzen Reihe von Bedrohungen ausgesetzt ist. Besonders auffällig ist, dass viele der Bedrohungen durch ein fehlendes Sicherheitsbewußtsein des Benutzers entstehen. Dies ist gerade im gewählten Szenario bedenklich, da davon ausgegangen werden kann, dass eine Schulung bzw. Sensibilisierung der Benutzer nicht im erforderlichen Umfang durchgeführt werden kann. Im Szenario ist außerdem davon auszugehen, dass sich die Benutzer nicht auf kompliziert anzuwendende Sicherheitsmaßnahmen einlassen werden. Diese Tatsache sollte im Entwurf unbedingt berücksichtigt werden.

Viele Bedrohungen beruhen darauf, dass ein Angreifer die Identität eines Benutzers übernimmt. In wie weit dies Auswirkungen hat, hängt von den Rechten des jeweiligen Benutzers, sowie von den Daten ab, die auf dem PDA gespeichert sind. Um das Gefahrenpotential zu reduzieren, sollten möglichst keine bzw. wie in (Karygiannis und Owens, 2002) gefordert,

[3] Mit Bugs werden Softwarefehler bezeichnet.

möglichst wenig sensible Daten auf dem mobilen Endgerät selbst gespeichert werden. Allerdings muss bei der Auslagerung von Daten auf den Server in Kauf genommen werden, dass dem Benutzer die Daten nur dann zur Verfügung stehen, wenn das mobile Endgerät eine Verbindung zum Server herstellen kann.

5 Entwurf

5.1 Annahmen und Eingrenzungen

Der Entwurf findet unter der Annahme statt, dass ein mobiles Endgerät zur Verfügung steht, welches die nachfolgend Eigenschaften erfüllt.

Das verwendete Betriebssystem ist ein Einbenutzerbetriebssystem ohne Rechteverwaltung. Diese Einschränkung wurde vorgenommen, da die meisten der bisher verfügbaren Betriebssysteme für mobile Endgeräte wie Windows CE (Pocket PC), EPOC und PalmOS dieser Anforderung entsprechen. Dieser Umstand wird sich auch, nach Auskunft eines Microsoft Mitarbeiters, zumindest für den Pocket PC in absehbarer Zeit nicht ändern. Dies ist auch der Grund, warum außerdem davon ausgegangen wird, dass ein PDA jeweils nur einer Person zugeordnet ist. Der Entwurf berücksichtigt damit nicht, wie beispielsweise ein PDA von einer gesamten Familie genutzt werden kann.

Des Weiteren muss der PDA und der Server in der Lage sein, eine TCP/IP Verbindungen über das drahtlose Netzwerk herzustellen. Beim Kommunikationskanal wird davon ausgegangen, dass dieser über Funk hergestellt wird. Über den den verwendeten Access Point und Server werden keine weiteren Annahmen getroffen.

5.2 Maßnahmen

Anhand der beschriebenen Sicherheitsprinzipien und der festgestellten Bedrohungen aus Kapitel 4.3 können nun geeignete Maßnahmen getroffen werden, um das System sicher zu machen. Die einzelnen Maßnahmen werden dabei den einzelnen Angriffszielen aus Kapitel 4.3 zugeordnet, wobei vor allem die Blätter der einzelnen Bedrohungsbäume interessant sind. Sollte es jedoch Maßnahmen geben, die für ganze Teilbäume sinnvoll sind, so werden diese in dem jeweiligen Wurzelknoten vorgestellt. Bei den beschriebenen Maßnahmen stehen wiederum technische Aspekte im Vordergrund.

5.2.1 PDA

p.1 Benutzeridentität vortäuschen

Um das Vortäuschen der Benutzeridentität zu erschweren, sollte nicht nur eine lokale Authentifizierung, sondern auch eine Authentifizierung bei der Inanspruchnahme von kritischen Diensten erfolgen. Dieses Authentifizierungsverfahren sollte sich möglichst vom lokalen Verfahren unterscheiden. Ansonsten ist es wahrscheinlich, dass mit der Überwindung des einen Anmeldeverfahrens das andere ebenfalls umgangen werden kann. Ein solches Vorgehen wird nach (Russel und Gangemi, 1991, S. 243) auch Zwei-Faktoren Authentifizierung[1] genannt. Es sollte jedoch nicht vergessen werden, dass dies die Benutzerfreundlichkeit herabsetzt. In wie weit dieses Vorgehen akzeptabel ist, hängt stark vom jeweiligen Szenario und den angebotenen kritischen Diensten ab.

p.1.1.1 Gerät aneignen

Es gibt keinen für den Benutzer vertretbaren Schutz vor Diebstahl. Eine Befestigung des Gerätes am Benutzer ist weder praktikabel, noch ist damit zu rechnen, dass ein solches Verfahren vom Gast akzeptiert wird. Deshalb sollte versucht werden, den Diebstahl des Gerätes so unattraktiv wie möglich zu machen bzw. den Schaden, der durch einen Diebstahl entstehen könnte, zu minimieren.

Um die Daten des Gerätes zu schützen, bzw. den Nutzen eines gestohlenen Gerätes zu reduzieren, sollte, wie im Beispielszenario beschrieben, nach dem Einschalten des mobilen Endgerätes eine Authentifizierung erforderlich sein. Der PDA sollte dazu nicht, wie in (Dedo, 2002) empfohlen, bereits beim Einschalten Informationen über den Benutzer preisgeben. Stattdessen sollte das Gerät mit dem Namen des Clubs sowie dessen Anschrift und Telefonnummer versehen werden, um die Rückgabe von gefundenen Geräten zu beschleunigen (Karygiannis und Owens, 2002, 5-32). Die Preisgabe von Benutzerinformationen ist fragwürdig, da nach (Schneier, 2000) in Zukunft Angriffe vermehrt auf Personen zielen werden und demnach solche Hinweise für einen Angreifer sehr nützlich sein können. Dies ist insbesondere dann der Fall, wenn ein wissensbasiertes Authentifizierungsverfahren zum Einsatz kommt, bei dem der Benutzer selbst die Zugangsdaten wählen kann. In Abschnitt p.1.1.2.1 wird auf diesen Sachverhalt näher eingegangen.

Um den Nutzen der Überwindung der lokalen Authentifizierung zu minimieren sollte die serverseitige Deaktivierung des Accounts so früh wie möglich nach dem Abhandenkommen des Gerätes erfolgen. Dazu ist es notwendig, dass dem Gast die Wichtigkeit dieser Maßnahme vermittelt wird. Außerdem sollte dem Benutzer klar gemacht werden, dass er

[1] two-factor authentication

das mobile Endgerät nicht weitergeben bzw. unbeaufsichtigt liegen lassen sollte. Zusätzlich sollte er über die durch ein Abhandenkommen des Gerätes entstehenden Gefahren aufgeklärt werden.

p.1.1.2.1 Verfahrensschwächen ausnutzen

Wie bereits in der Analyse beschrieben wurde, kommt es nicht nur auf das reine Vorhandensein eines Authentifizierungsverfahrens an, sondern vor allem auf dessen Qualität. Dabei können je nach verwendetem Verfahren unterschiedliche Kriterien angewendet werden. Da es eine sehr große Anzahl verschiedener Authentifizierungsverfahren gibt, soll hier nur kurz anhand eines Passwortmechanismus gezeigt werden, wie die Qualität bewertet werden kann.

Zum einen darf das Passwort nicht zu einfach sein, damit es nicht erraten, oder durch Brute-Force-Attacken ermittelt werden kann. Zu einfache Passwörter sollten demnach vom System selbst zurückgewiesen werden. Das Passwort darf jedoch auch nicht zu kompliziert sein, da dies sonst den Benutzer zu sehr belastet. Dieser könnte daraufhin dazu übergehen, sich die benötigten Daten aufzuschreiben. Das wiederum bietet neue Angriffsmöglichkeiten. Eine Umfrage (CentralNic, 2001) unter 1200 Personen an 30 Firmen hat außerdem ergeben, dass etwa 47,5% Familien, 32% Fan und 11% Fantasie basierte Passwörter wählen. Bei 55,3% der familienbasierten Passwörter handelt es sich der Umfrage nach um den Benutzernamen oder einer Variation davon. Liegt dem lokalen Zugangskontrollverfahren ein vom Gast selbst gewähltes Passwort zugrunde, so lassen sich demnach mit hoher Wahrscheinlichkeit die Zugangsdaten erraten. Ähnliche Ergebnisse werden auch in einer anderen Quelle genannt (Tanenbaum, 2001).

Sollte es sich nicht wie im gewählten Szenario um eine Umgebung handeln, in der sich die Benutzer nur relativ kurz aufhalten, sollte außerdem dafür gesorgt werden können, dass die Daten nach einer gewissen Zeitspanne geändert werden müssen. Des weiteren sollte die Eingabe des Passworts so geschehen, dass ein Beobachter diese nicht mitlesen kann. Demnach sollte das eingegebene Passwort niemals im Klartext angezeigt werden. Ein weiteres Qualitätsmerkmal ist die Speicherung der Daten im System selbst. Wird das Passwort beispielsweise als Klartext in einer Datei gespeichert, so ist das Verfahren wesentlich angreifbarer als ein anderes, welches nur einen Einweg-Hashwert[2] des Passwortes speichert. Um einen zusätzlichen Schutz zu gewähren, sollte außerdem dafür gesorgt werden, dass dem Benutzer nur eine begrenzte Anzahl von fehlgeschlagenen Authentifizierungsversuchen gewährt werden. Sollte auch die letzte Authentifizierung nicht erfolgreich sein, so sollte das Gerät gesperrt und alle schützenswerten Daten gelöscht werden.

[2] Ein Einweg-Hashwert ist das Ergebnis einer Einweg-Hashfunktion. Eine solche Funktion errechnet nach (Wobst, 1997) aus Eingabedaten einen komprimierten Wert. Es wird außerdem gefordert, daß die Funktion für verschiedene Daten mit ausreichend großer Wahrscheinlichkeit verschiedene Werte erzeugt. Zusätzlich sollte es mit vernünftigem Aufwand nicht möglich sein, eine Bytefolgen zu konstruieren, die einen vorgegebenen Wert ergibt.

Es lässt sich leicht ersehen, wie vielfältig die Kriterien für die Qualität eines Verfahrens ausfallen können. Eine gute Übersicht verschiedener Authentifizierungsverfahren für PDAs bietet (Jansen, 2003).

p.1.1.2.2 Speicher analysieren

Um ein Analysieren des Speichers zu verhindern, sollten die Authentifizierungsdaten grundsätzlich nur in verschlüsselter Form gespeichert werden. Besser wäre es jedoch, wenn nicht die Daten an sich, wie zum Beispiel das Passwort, sondern nur die durch eine Einweg-Hashfunktion generierten Werte gespeichert werden. Beispiel einer solchen Einweg-Hashfunktion ist der secure-hash-algorithm (SHA), welcher aus einer Zusammenarbeit des NIST (National Institute of Standards and Technology)[3] und der NSA (National Security Agency)[4] entwickelt wurde.

p.1.1.2.3 Authentifizierungsdaten verändern

Methoden, die der Bedrohung durch Veränderungen von Daten und Programmen entgegenwirken, werden in Abschnitt p.2 vorgestellt.

p.1.1.2.4 Hard Reset

Um die Gefährdungen durch einen Hard Reset zu minimieren, sollte dafür gesorgt werden, dass dabei möglichst viele Daten gelöscht werden. Insbesondere Personen und Firmen bezogene Programme und Daten sollten gelöscht werden. Dies steht zwar im Widerspruch zu den von Microsoft verfassten Sicherheitsrichtlinien (Dedo, 2002), aber die Durchführung eines Hard Resets sollte meiner Meinung nach immer einen Ausnahmefall darstellen. Deshalb ist ein Mehraufwand durch das Neuinstallieren der Software aus Sicherheitsgründen durchaus vertretbar. Des Weiteren wäre es leichtsinnig, den Daten eines wiedergefundenen PDAs zu vertrauen. Generell sollten wiedergefundene PDAs, immer komplett neu installiert werden. Dazu bieten sich das Aufspielen vorkonfigurierter Images an, die die sichere Konfiguration der PDAs erheblich vereinfachen.

p.1.2 Gerät im authentifizierten Zustand entwenden

Der PDA sollte so eingestellt sein, dass er sich nach einer gewissen Zeit, in der der Benutzer untätig ist, selbst abschaltet. Außerdem sollten, wie bereits erwähnt, alle kritischen

[3] http://www.nist.gov
[4] http://www.nsa.gov

Dienste eine erneute Authentifizierung erfordern. Des Weiteren sollte der Benutzer über diese recht große Gefahr informiert werden.

p.1.3 Böswillige Programme einspielen

Sollte das Einspielen fremder Programme nicht komplett verhindert werden können, sollte auf dem mobilen Endgerät ein Virenscanner installiert werden. Außerdem sollte dafür gesorgt werden, dass sich der Virenscanner selbst aktualisiert bzw. der Benutzer gegebenenfalls darauf hingewiesen wird, dass das Programm vom PDA-Betreuer aktualisiert werden muss.

Zusätzlich kann eine Firewall auf dem mobilen Endgerät installiert werden, die nur Verbindungen zu vertrauenswürdigen Servern zulässt. Die Konfiguration der Firewall sollte sich durch den Benutzer nicht ändern lassen.

Weitere Schutzmaßnahmen gegen solche Angriffe finden sich in Abschnitt p.3.

p.1.4 Clientanwendung vortäuschen

Um zu gewährleisten, dass der Server vorgetäuschte Clientanwendungen erkennt, sollten sich echte beim Verbindungsaufbau dem Server gegenüber authentifizieren.

Zusätzlich bietet die bereits angesprochene Benutzerauthentifizierung vor der Inanspruchnahme kritischer Dienste einen weiteren Schutz vor Angreifern, welche versuchen, die Clientanwendung zu emulieren.

p.1.4.1 Datenverkehr analysieren

Maßnahmen hierzu sind in Abschnitt c.2 beschrieben.

p.1.4.2 Clientanwendung analysieren

Das Analysieren der Clientanwendung lässt sich nicht ohne weiteres verhindern. Sobald ein Angreifer lesend auf ein Programm zugreifen kann, ist er in der Lage, dieses zu analysieren. Ziel muss es demnach sein, die Analyse so weit wie möglich zu erschweren. Dazu können so genannte Obfuscator eingesetzt werden, die den Programmablauf durch Verkomplizierung verschleiern, aber die eigentliche Funktionsweise erhalten. Es ist die nach

(Collberg und Thomborson, 2002) wohl einzige Möglichkeit, das Reverse Engineering effektiv zu erschweren. Soweit es möglich ist, sollte jedoch dem Kerkhoffschen Prinzip[5] gefolgt werden.

Außerdem sollten alle Anwendungen, die das Auslesen der Programmdaten ermöglichen, gelöscht, und das Einspielen fremder Software auf das mobile Endgerät verhindert werden.

Weiter Maßnahmen befinden sich in Abschnitt p.2.

p.1.4.3 Clientanwendung kopieren

Um das Kopieren der Clientanwendung zu verhindern, sollten alle nicht benötigten Kommunikationskanäle gesperrt werden. Ausgenommen bleibt der vom Clientprogramm benutzte Kanal zur Kommunikation mit dem Dienstserver. Des Weiteren sollte verhindert werden, dass schreibend auf unbekannte externe Speicherkarten zugegriffen werden kann. Außerdem sollten alle Programme gelöscht werden, mit denen es möglich wäre, Daten zu kopieren.

p.2 Programme oder Daten verändern

Das Verändern von Daten lässt sich nicht generell verbieten, da viele Programme (Adressverwaltung, Terminverwaltung, etc.) die eigenen Daten auf dem Gerät speichern und verändern müssen. Da es nach Kapitel 5.1 keine Rechteverwaltung auf Benutzerebene gibt, müssen neue Wege gefunden werden, diese Daten sicher zu verwahren.

p.2.1.1 Gerät aneignen

Wie bereits erwähnt, gibt es für diese Bedrohung keinen direkten Schutz. Maßnahmen, die diese Gefahr dennoch reduzieren, wurden im vorangegangen Abschnitt p.1.1.1 vorgestellt.

[5] Das Kerkhoffsche Prinzip besagt, dass die Sicherheit eines Systems nicht auf der Geheimhaltung der angewendeten Verfahren und Mechanismen beruhen darf, da davon ausgegangen werden kann, dass sich ein Angreifer diese Informationen beschaffen kann (Kerckhoff, 1883).

p.2.1.2.1 Zugangskontrollverfahren überwinden

Sollte der Angreifer erfolgreich die Identität des Benutzers vortäuschen können, so hat dieser die gleichen Rechte wie der Gast, dessen Identität angenommen worden ist. Er ist also in der Lage alle Programme auszuführen und anzuwenden. Deshalb sollten alle Programme gelöscht werden, mit denen es potenziell möglich ist, beliebige Dateien zu editieren. Sollte es dem Angreifer jedoch möglich sein, fremde Programme auf das mobile Endgerät zu übertragen, müssen weitere Maßnahmen zur Absicherung der Daten und Programme getroffen werden.

Die einfachste Methode, um das Verändern von Programmen zu verhindern, ist die Hinterlegung auf einem ROM-Speicher. Dies hat jedoch zwei gravierende Nachteile. Zum einen gestaltet sich das Aktualisieren der Programme als schwierig und kostspielig. Zum anderen können diese Daten nicht, wie unter p.1.1.2.4 gefordert, bei einem Hard Reset gelöscht werden.

Eine weitere Möglichkeit besteht in der Bildung und Speicherung von Hashsummen. Dabei wird durch eine Einweg-Hashfunktion für jedes Programm ein Hashwert aus dessen Bits gebildet. Diese kryptographische Prüfsumme wird, für den Benutzer unzugänglich, gespeichert. Eine Betriebssystemkomponente kann daraufhin, bevor ein Programm gestartet wird, überprüfen, ob der derzeitige Hashwert noch mit dem gespeicherten Wert übereinstimmt. Sollten sich beide Werte unterscheiden, so wurden Änderungen am Programm vorgenommen. In einem solchen Fall muss die Ausführung des Programms sofort abgebrochen werden. Diese Lösung ist jedoch nicht hundertprozentig sicher, da ein Angreifer ein Programm ändern und selbst einen neuen passenden Hashwert erzeugen kann.

Sicherer ist es, wenn der PDA-Betreuer sämtliche Programme digital signiert und diese Signaturen auf dem PDA hinterlegt werden. Zusätzlich muss der öffentliche Schlüssel des PDA-Betreuers auf dem PDA hinterlegt werden, damit das Betriebssystem diesen zur Überprüfung der Signaturen verwenden kann. Um zu verhindern, dass ein Angreifer diesen ersetzt, sollte der Schlüssel in einem Speicher aufbewahrt werden, auf den nur lesend zugegriffen werden kann. Die oben beschriebenen Nachteile kommen hierbei kaum zu tragen, da sich der öffentliche Schlüssel des PDA-Betreuers nur sehr selten ändert und der öffentliche Schlüssel an sich nicht schützenswert ist.

Ist die Bildung von Signaturen nicht möglich, so sollte wenigstens ein Virenscanner, sowie eine Firewall auf dem Handheld installiert werden.

In den vorangegangenen Absätzen wurde nur die Sicherheit von Programmen, nicht jedoch die Sicherheit von Anwendungsdaten, diskutiert. Das Verhindern von böswilligen Datenveränderungen lässt sich wesentlich schwieriger verhindern. Zwar kann auch hier mit digitalen Signaturen gearbeitet werden, doch müsste dazu auch der private Schlüssel auf dem Gerät hinterlegt werden. Dies liegt daran, dass bei jeder Änderung des Datenbestandes eine neue Signatur erstellt werden muss. Ein solches Vorgehen ist recht zeitaufwendig und damit nicht benutzerfreundlich. An dieser Stelle muss also je nach Szenario überlegt werden,

was für Daten auf dem mobilen Endgerät gespeichert werden und wie schützenswert diese sind. Da im gewählten Szenario keine kritischen Daten auf dem mobilen Endgerät gespeichert werden müssen, wird hier nicht weiter darauf eingegangen.

Alle Daten, die der Benutzer nicht verändern sollte bzw. darf, können wie Programme vom PDA-Betreuer signiert werden. Dies ist insbesondere für sicherheitskritische Veränderungen an Betriebssystemeinstellungen interessant.

p.2.1.2.2 Speicher direkt editieren

Es kann nicht verhindert werden, dass ein Angreifer versucht, den Speicher des Gerätes direkt zu editieren. Deshalb können nur Maßnahmen getroffen werden, die solche Änderungen erkennen oder aber erschweren.

Neben den unter p.2.1.2.1 beschriebenen Schritten, können solche Angriffe relativ einfach dadurch erschwert werden, dass der gesamte Speicher des mobilen Endgerätes verschlüsselt wird. Die Entschlüsselung sollte dabei an eine erfolgreich verlaufende lokale Authentifizierung gebunden sein, oder für alle wichtigen Daten gesondert durchgeführt werden. Dabei hängt es wieder vom jeweiligen Szenario ab, was den Benutzern zugemutet werden kann und was nicht.

p.2.2 Fernzugriff auf Speicher

Um zu verhindern, dass ein Angreifer aus der Ferne Zugriff auf das Gerät erhält, sollten grundsätzlich alle Kommunikationskanäle gesperrt werden, die nicht unbedingt benötigt werden. Dies kann durch die Verwendung von Firewalls geschehen. Die Konfiguration der Firewall sollte vom Benutzer nicht verändert werden können. Beim Einsatz einer Firewall sollte jedoch unbedingt darauf geachtet werden, ob die Ressourcen des mobilen Endgerätes ausreichen, um dem Benutzer noch ein erträgliches Arbeiten mit dem Gerät zu ermöglichen. Auf den Einsatz einer Firewall kann verzichtet werden, wenn es gelingt, das Einspielen fremder Software und das Modifizieren der bestehenden Software zu verhindern.

p.3 Externe Programme einspielen

Die wohl einfachste Lösung wäre auch hier, alle Programme auf ROM-Speichern zu hinterlegen. Die Daten der Programme könnten weiter auf beschreibbaren Speichern lagern. Das Betriebssystem muss dann dafür sorgen, dass nur Programme vom verwendeten ROM-Speicher ausgeführt werden können. Diese Lösung ist jedoch, wie unter p.2.1.2.1 bereits erwähnt, unpraktikabel.

Der bereits angesprochene Ansatz der Verwendung von digitalen Signaturen erweist sich hier ebenfalls als nützlich. Da das Betriebssystem demnach nur Anwendungen ausführt, zu denen es eine gültige Signatur gibt, wäre über diesen Weg auch das Ausführen extern eingespielter Programme ohne entsprechende Signatur nicht mehr möglich.

p.3.1 Speicher direkt editieren

Maßnahmen, die sich mit dieser Bedrohung beschäftigen, wurden bereits in Kapitel p.2.1.2.2 vorgestellt.

p.3.2 Speichermodule einbauen

Das Betriebssystem sollte es dem PDA-Betreuer ermöglichen, für alle Speichererweiterungsschächte Sicherheitsrichtlinien zu definieren. Es sollte darüber möglich sein, dem Anwender das Arbeiten mit neu hinzugefügten Speichermodulen zu verbieten. Sollte das Betriebssystem diese Funktionalität nicht unterstützen, so bietet sich die Verwendung von Hardwaresperren an, wie es sie schon seit längerer Zeit für Diskettenlaufwerke gibt.

p.3.3 Kommunikationskanäle nutzen

Um das Einspielen externer Software über Kommunikationskanäle zu verhindern, sollten, wie in Abschnitt p.2.2 beschrieben, grundsätzlich alle vom PDA bereitgestellten Kommunikationskanäle gesperrt werden. Es sollten nur vordefinierte Verbindungen zu den vom Ferienclub betriebenen Servern zugelassen werden. Außerdem sollten alle Programme gelöscht werden, mit denen potenziell Dateien übertragen werden könnten. Dies beinhaltet insbesondere Anwendungen wie Browser oder FTP-Clients.

Sollte es nicht verhindern werden können, dass externe Programme eingespielt werden, so sollte unbedingt eine Firewall und ein Virenscanner auf dem PDA installiert werden.

Außerdem sollte verhindert werden, dass der Gast die gemachten Einstellungen verändern kann. Dies ließe sich wieder über die bereits besprochenen Signaturen gewährleisten.

5.2.2 Kommunikationskanal

c.1 Verfügbarkeit

c.1.1 Störsender

Da der Erfolg eines solchen Angriffs von der gewählten Übertragungstechnologie abhängt, lässt sich keine generelle Maßnahme zur Abwehr solcher Angriffe treffen. Es sollte deshalb bei der Auswahl der Technologie darauf geachtet werden, wie diese mit dem Problem umgeht und welche Möglichkeiten sie bietet, das Problem zu vermeiden bzw. zu minimieren.

c.2 Vertraulichkeit

c.2.1 Kommunikationskanal abhören

Durch eine geschickte Platzierung der Access Points können solche Attacken bereits stark eingeschränkt werden. Dazu sollte eine Felduntersuchung gemacht werden, die durch entsprechende Tools unterstützt werden kann. Ziel dieser Untersuchung ist es unter anderem eine optimale Platzierung der Access Points zu erarbeiten, so dass das dadurch aufgespannte Netz möglichst nicht über die dafür vorgesehenen Grenzen reicht. Weitere Informationen über die Durchführung einer solchen Untersuchung finden sich bei (Geier, 2002).

Zusätzlich sollten alle Daten, die über das drahtlose Netz geschickt werden, verschlüsselt werden. Dabei ist es wichtig, dass sowohl Client als auch Server das Verschlüsselungsprotokoll in einer sicheren Konfiguration anwenden. Dies ist deshalb wichtig, da einige Implementierungen durch falsche Konfiguration anfällig gemacht werden können[6].

c.2.2 Man-in-the-Middle-Attacke

Diese Art der Bedrohung lässt sich durch geeignete Wahl eines kryptographischen Kommunikationsprotokolls verhindern. In Protokollen, die mit asymmetrischen Verschlüsselungsverfahren arbeiten, wie beispielsweise SSL (Secure Sockets Layer) bzw. TLS (Transport Layer Security), kann dies durch die Speicherung der echten, gültigen öffentlichen Schlüssel der Kommunikationspartner gewährleistet werden.

[6] Die VPN (Virtual Private Network) Konfiguration von Windows XP lässt es beispielsweise zu, dass aufgrund verschiedener Umstände statt der eingestellten IPSec (IP Security) die unsichere PPTP (Point-to-Point Tunneling Protocol) Verschlüsselung verwendet wird (Ornaghi und Valleri, 2003)

c.3 Integrität

c.3.1 Man-in-the-Middle-Attacke

Maßnahmen zur Verhinderung eines solchen Angriffs wurden bereits in Abschnitt c.2.2 dargestellt. Deshalb soll hier nicht erneut darauf eingegangen werden.

c.3.1 Wiedereinspielen von Nachrichten

Um zu verhindern, dass das Wiedereinspielen von abgehörten Nachrichten keinen Schaden anrichtet, sollte ein Kommunikationsprotokoll gewählt werden, welches solche Angriffe erfolgreich verhindern kann. Protokolle gehen dabei oft so vor, dass sie neben den reinen Daten zusätzlich Sequenznummern und/oder Sitzungsschlüsseln verwenden.

5.2.3 Access Point und Server

Serveranwendung vortäuschen

Um zu verhindern, dass ein Angreifer dem Client die Serveranwendung vortäuschen kann, sollte sich der Server dem Client gegenüber authentifizieren.

Generell bietet sich für die gegenseitig geforderte Authentifizierung ein Verfahren an, bei dem auf dem Client so wenig verräterische Informationen wie möglich gespeichert werden.

5.2.4 Sonstige Angriffe

Bugs

Leider gibt es keine Möglichkeit, um einwandfrei festzustellen, ob eine Software frei von Fehlern ist. Deshalb bietet sich hier die Berücksichtigung erprobter Maßnahmen an, um die Fehleranfälligkeit von Systemen systematisch zu reduzieren. Bei dem Einsatz von Fremdsoftware sollte man deshalb unbedingt darauf achten, dass diese regelmäßig gepatched und auf dem neusten Stand gehalten wird (Allen, 2001). Außerdem sollten nur die Dienste vorhanden sein, die für das Szenario zwingend erforderlich sind. Bei der Entwicklung eigener Software sollten die Entwickler unbedingt für das Thema Sicherheit sensibilisiert werden. Gerade in diesem Bereich gibt es viele lesenswerte Bücher[7] und Internetseiten[8].

[7] (Howard und LeBlanc, 2002),(Graff und Wyk, 2003), etc.
[8] http://www.owasp.org/, http://www.secureprogramming.com/, etc.

Sollten die Benutzer das mobile Endgeräte über einen längeren Zeitraum verwenden, muss außerdem die Möglichkeit geschaffen werden, Softwareupdates auf das mobile Endgerät zu übertragen.

Abstreitbarkeit

Um das Abstreiten der Nutzung von Diensten zu verhindern, bietet sich der Einsatz von digitalen Signaturen an. Dies ist insbesondere deshalb sinnvoll, da mit dem 2001 in Kraft getretenen Signaturgesetz die Rechtsverbindlichkeit solcher Signaturen durchgesetzt worden ist. Im gewählten Szenario sollten demzufolge alle kritischen Dienstanfragen vom Client signiert werden. Da der Zugriff auf den privaten Schlüssel immer eine erneute Authentifizierung erfordern sollte, kann diese ebenfalls für die bereits angesprochene Dienstauthentifizierung benutzt werden. Dies ist sinnvoll, da beide dem gleichen Zweck dienen: Der Authentifizierung des Benutzers.

5.2.5 Organisatorische Maßnahmen

Bevor auf die einzelnen Bedrohungen eingegangen wird, soll an dieser Stelle eine wichtige organisatorische Maßnahme beschrieben werden, die das Gefahrenpotenzial der Veränderung der Abrechnungsdaten wesentlich verringert. Die verursachbaren Kosten pro Gast sollten unbedingt limitiert werden. Dies kann entweder durch eine feste maximale Grenze realisiert werden, oder besser, durch einen vom Gast wählbaren Betrag, den dieser im Vorfeld einzahlen muss. Kostenpflichtige Dienste können dann direkt mit dem Guthaben des Gastes verrechnet werden.

5.2.6 Zusammenfassung

In den vorangegangenen Abschnitten wurden viele verschiedene Maßnahmen vorgestellt,um das in Kapitel 2 vorgestellte System abzusichern.

Aus Gründen der Übersichtlichkeit zeigt Tabelle 5.1 noch einmal eine Liste aller Maßnahmen, sowie die Bedrohungen, denen diese entgegenwirken. Anhand der Tabelle läßt sich ebenfalls erkennen, dass alle Bedrohungen aus Kapitel 4.3 berücksichtigt worden sind. Viele der Lösungen sind dabei so gewählt, dass sie transparent für den Benutzer sind. Dies bedeutet, dass dieser durch die getroffenen Sicherheitsmaßnahmen nicht belastet wird. Einzige Ausnahme bildet die Dienstauthentifizierung, deren Sinn sich jedoch dem Benutzer leicht vermitteln lassen sollte.

Es sei nochmal darauf hingewiesen, dass nicht alle Maßnahmen umgesetzt werden müssen. Einige Maßnahmen sind nur dann sinnvoll, wenn sich andere z.B. auf Grund der eingesetzten Komponenten nicht realisieren lassen. So ist beispielsweise der Einsatz eines

Maßnahme	Bedrohungen
Dienstauthentifizierung	p.1, p.1.4
lokale Authentifizierung	p.1.1.1
Deaktivierung des Accounts	p.1.1.1
sicherer Passwortspeicher	p.1.1.2.2
Speicherverschlüsselung	p.1.1.2.3, p.2.1.2.2
Gerät löschen bei Hard Reset	p.1.1.2.4
Bereitstellung von Images	p.1.1.2.4
automatisch ausloggen	p.1.2
Virenscanner	p.1.3
Firewall	p.1.3, p.2.2
Clientauthentifizierung	p.1.4
Obfuscating	p.1.4.2
Überflüssige Programme löschen	p.1.4.3
externe Speicherkarten verbieten	p.1.4.3, p.3.2
Programmsignaturen prüfen	p.2.1.2.1, p.3
Konfigurationsveränderungen verbieten	p.2.1.2.1
überflüssige Kanäle sperren	p.1.1.2.3, p.2.2, p.3.3
sichere Kanalverschlüsselung	c.2, c.3
Serverauthentifizierung	Serveranwendung vortäuschen
Remoteupdate	Bugs
Digitale Signatur	Abstreitbarkeit

Tabelle 5.1: Entwurf: Maßnahmenkatalog

Virenscanners auf dem mobilen Endgerät nur dann sinnvoll, wenn nicht verhindert werden kann, dass bestehende Programme verändert oder unbekannte ausgeführt werden können. Welche Maßnahmen davon betroffen sind, wurde in den einzelnen Abschnitten diskutiert.

In Bezug auf die Bedrohungsanalyse lässt sich feststellen, dass die Einschränkung auf Einbenutzerbetriebssysteme ohne Rechteverwaltung dazu geführt hat, dass sich einige der Hauptbedrohungen nur schwierig verhindern ließen. In diesem Zusammenhang sind eine ganze Reihe von Maßnahmen entstanden, um speziell den Bedrohungen, die durch die Übernahme der Benutzeridentität entstehen, entgegenzuwirken. Ein Großteil dieser Maßnahmen lässt sich allerdings nur auf Betriebssystemebene realisieren.

Die Forderung nach einer Auslagerung sensibler Daten auf den Server wurde bisher nicht berücksichtigt. Dies liegt vor allem daran, dass ein Großteil der analysierten Daten anwendungsbezogen ist. Deshalb wird die Verteilung der Daten erst im nächsten Kapitel näher betrachtet.

5.3 Softwareentwurf

In diesem Abschnitt wird aufbauend auf den erarbeiteten Sicherheitsmaßnahmen der Entwurf einer sicheren Clientanwendung erfolgen, mit der sich sowohl lokale als auch Netzwerkdienste nutzen lassen. Maßnahmen, die sich nur auf Betriebssystemebene umsetzen lassen, werden hier nicht näher betrachtet, da deren Realisierung Aufgabe der einzelnen Hersteller ist. Demnach beschränkt sich der Cliententwurf auf die folgenden Maßnahmen:

1. Dienstauthentifizierung
2. Clientauthentifizierung
3. Programmsignaturen prüfen
4. sichere Kanalverschlüsselung
5. Serverauthentifizierung
6. Remoteupdate
7. Digitale Signatur

Für die Authentifizierung wurde dabei ein PKI (Public Key Infrastructure) basierter Ansatz gewählt. Dies hatte mehrere Gründe. Zum einen ist für die Erstellung von digitalen Signaturen ohnehin das Vorhandensein eines öffentlichen und privaten Schlüssels erforderlich. Zum anderen müssen dadurch für die Serverauthentifizierung, wie in Abschnitt 5.2.3 gefordert, keine sensitiven Informationen auf dem mobilen Endgerät gespeichert werden. Der Client muß für die Authentifizierung nur im Besitz des öffentlichen Schlüssels des Servers sein. Weitere Informationen zum Thema PKI und Zertifikate finden sich in (Adams und Lloyd, 2002).

Als Zertifikatstyp wurden X509 V3 Zertifikate gewählt. Diese Art der Zertifikate hat den Vorteil, dass sie bereits weit verbreitet ist, und dass sie sich über so genannte Extensions erweitern lässt. Die Verwendung der Extensions kann besonders dann interessant werden, wenn Gäste mit verschiedenen Rechten ausgestattet werden sollen. Auch die gezielte Bereitstellung von Diensten für andere Rollen lässt sich dadurch leicht realisieren. Für weitere Informationen sei auch hier auf (Adams und Lloyd, 2002) verwiesen.

5.3.1 Vorüberlegungen

Bei der Betrachtung der einzelnen Problempunkte fällt auf, dass es im Wesentlichen um vier Bereiche geht.

Der erste Bereich beschäftigt sich mit der Verwaltung der verschiedenen Zertifikate und privaten Schlüssel, die für die Clientanwendung benötigt werden.

Der zweite Bereich betrifft den Aufbau eines verschlüsselten Kommunikationskanals zum Dienstserver und die dazugehörige Client- und Serverauthentifizierung.

Ein weiterer Teil dient der Erstellung und Überprüfung von digitalen Signaturen.

Der letzte Bereich beschäftigt sich mit der Darstellung von Dienstanfragen und -antworten. Diese sollte sowohl die Inanspruchnahme von normalen und kritischen Diensten unterstützen, als auch das Senden von Updates an das mobile Endgerät.

Es bietet sich demnach an, die gesamte Applikation in vier Module zu zerlegen. Bevor die einzelnen Module genau beschrieben werden, ist es jedoch sinnvoll, über die Verteilung der einzelnen Datenarten aus Kapitel 4.2 nachzudenken.

5.3.2 Verteilung der Daten

Hinsichtlich der Software muss entschieden werden, wie viel Logik und Daten sich auf dem Client, also dem mobilen Endgerät, und auf dem Server befinden. Dementsprechend spricht man von Fat- oder Thin-Client.

Wie bereits in der Bedrohungsanalyse festgestellt wurde, sollte ein möglichst großer Teil der vom Client benötigten Daten auf dem Server gespeichert werden, da dieser besser geschützt werden kann.

Abrechnungsdaten

Die Abrechnungsdaten lassen sich ohne weiteres serverseitig speichern. Ein Nachteil, der dadurch entsteht, ist, dass der Client zur Abfrage der bisher entstandenen Kosten eine Verbindung zum Netzwerk haben muss. Der Benutzer kann somit die bisher entstandenen Kosten nicht ohne weiteres einsehen. Es wäre jedoch möglich eine Kopie der Abrechnungsdaten auf dem Client zu speichern, die nur zur Kontrollanzeige für den Benutzer dienen. Da die Verletzung der Vertraulichkeit dieser Daten jedoch ebenfalls bedenklich sein kann, muss im Einzelfall entschieden werden, ob die zusätzliche Speicherung der Daten auf dem mobilen Endgerät sinnvoll ist oder nicht.

PDA-Anwendungsdaten

Die PDA-Anwendungsdaten lassen sich nur schwer auf den Server auslagern, da der Client immer über ein paar Funktionalitäten verfügen muss. Diese beinhalten u.a. den Verbindungsaufbau zum Server und das Senden von Dienstanfragen. Die Daten für die Anzeige ließen sich komplett auf den Server auslagern. Dieses ist jedoch nicht sinnvoll, da der Client in der Lage sein soll, lokale Dienste auch ohne den Kontakt zum Server auszuführen bzw. darzustellen. Es wäre also besser, jene Anzeigedaten auf den Server auszulagern, die

ohnehin nur während der Kommunikation mit diesem benötigt werden. Dies entspricht den demnach den Anzeigedaten der Netzwerkdienste. Dies hat mehrere Vorteile. Zum einen ist es nach der Bedrohungsanalyse sinnvoll so viele Daten wie möglich auf dem Server zu speichern, zum anderen spart die Speicherung der Anzeigedaten auf dem Server wertvollen Platz auf dem mobilen Endgerät. Hinzu kommt, dass es dadurch sehr einfach wird, neue Netzwerkdienste hinzuzufügen oder bestehende zu verändern, da die Clientsoftware dafür nicht angepasst werden muß. Dies hat eine bessere Wartbarkeit und Erweiterbarkeit zur Folge. Zwar erfordert das Empfangen und Interpretieren einige Rechenleistung vom mobilen Endgerät, aber WAP-fähige Mobiltelefone haben gezeigt, dass die Geschwindigkeit der Geräte für einen solchen Ansatz bereits durchaus ausreicht. Um gleichzeitig Portabilität zu gewährleisten, sollte außerdem eine plattformunabhängige Darstellung der Anzeigedaten gewählt werden. Beispiele hierfür sind XML (Extensible Markup Language)[9] basierte Ansätze wie XHTML (Extensible HyperText Markup Language)[10], XUL (XML User Interface Language)[11] oder Thinlets[12]. Weitere Informationen zu Thinlets folgen in Kapitel 6.

Das Senden von Codeteilen ist ebenfalls denkbar, doch ist dies oft nur sehr aufwendig zu realisieren. Während sich Anzeigedaten relativ leicht interpretieren lassen, ist das Interpretieren und Ausführen von Code recht aufwendig. Deshalb sollte je nach Einsatzgebiet ein solches Vorgehen gesondert geprüft werden. Der Entwurf sieht jedoch auch das Senden von Code vor.

Grundsätzlich müssen sich alle Informationen und Daten, die für die Nutzung lokaler Dienste erforderlich sind, auf dem mobilen Endgerät befinden. Die Informationen und Daten für Netzwerkdienste, sollten weitestgehend auf dem Server gespeichert werden.

PDA-Authentifizierungsdaten

Die PDA-Authentifizierungsdaten müssen auf dem PDA gespeichert werden, da es dem Benutzer sonst nicht möglich wäre, das mobile Endgerät außerhalb des Netzwerkes zu nutzen. Da die PDA-Authentifizierung jedoch Teil des Betriebssystems ist, werden diese Daten im Softwareentwurf nicht berücksichtigt.

Zustandsdaten

Die Zustandsdaten lassen sich auf dem Server speichern. Dies ist sogar ausgesprochen sinnvoll, da damit effektiv verhindert werden kann, dass ein Angreifer mit einem gefälschten Gerät Dienste in Anspruch nehmen kann. Ein Nachteil dieser Lösung ist jedoch, dass nicht verhindert werden kann, dass ein Angreifer die lokalen Anwendungen nutzt, auch wenn

[9] http://www.w3.org/XML/
[10] http://www.w3.org/TR/xhtml1/
[11] http://www.mozilla.org/projects/xul/
[12] http://www.thinlet.com/

der Account des Gastes deaktiviert wurde. Kompensieren ließe sich das Problem dadurch, dass bei einem Verbindungsaufbau ins Clubnetzwerk das Clientprogramm eine Löschung, sowie Sperrung des Gerätes verursachen kann, falls dabei festgestellt wird, dass der Benutzer deaktiviert wurde. Da das Szenario keine clientseitigen kritischen Dienste vorsieht, können die Zustandsdaten jedoch ohne weiteres auf dem Server hinterlegt werden. Des Weiteren werden nach der in Kapitel 3 durchgeführten Analyse Sperrungen nur dann vorgenommen, wenn das Gerät entweder abhanden gekommen ist oder es beim Auschecken an den Ferienclub zurückgegeben wird.

5.3.3 Zertifikatsverwaltung

Abbildung 5.1: Klassendiagramm: Zertifikatsmanager

Das Modul zur Verwaltung von Zertifikaten und Schüsseln muss im Wesentlichen zwei Methoden bereitstellen: getCertificateChain() und getPrivateKey(). Die Methode getCertificateChain() dient dazu, die gesamte Zertifikatskette für einen spezifizierten CertificateKey zurückzuliefern. Der CertificateKey stellt dabei Merkmale zur Verfügung, um das angeforderte Zertifikat eindeutig zu spezifizieren. Je nachdem, wie die Zertifikate gespeichert sind, kann es sich dabei z.B. um einen Alias oder um einen LDAP (Lightweight Directory Access Protocol)-Namen handeln.

Die zweite Methode getPrivateKey() dient dazu, den privaten Schlüssel für einen bestimmten CertificateKey zu besorgen. Da der private Schlüssel meistens durch ein Passwort geschützt ist, kann dieses ebenfalls an die Methode übergeben werden.

Folgende Zertifikate sind vorgesehen:

1. PDA-Betreuer - Der öffentliche Schlüssel dieses Zertifikats wird benötigt, um die eigene Signatur des Programms zu überprüfen. Es sollte jedoch nicht vergessen werden, dass eine solche Überprüfung auf Betriebssystemebene wesentlich besser aufgehoben wäre. Der Softwareentwurf sieht die Prüfung der eigenen Anwendungssignatur dennoch vor, da viele bisher verfügbare Betriebssysteme eine solche Überprüfung nicht bieten. Des Weiteren wird damit wenigstens ein gewisser Schutz vor böswilligen Veränderungen geboten.

2. Client - Das Clientzertifikat dient zusammen mit dem dazugehörigen privaten Schlüssel dazu den PDA bzw. die Clientanwendung zu authentifizieren und um den verschlüsselten Kommunikationskanal aufzubauen.
3. Gast - Vom Benutzer wird ebenfalls der private Schlüssel benötigt, da dieser dazu dient, kritische Dienstanfragen zu signieren.
4. Server - Die Clientsoftware benötigt außerdem das Zertifikat mit dem öffentlichen Schlüssel des Servers, um diesen auf seine Vertrauenswürdigkeit beim Verbindungsaufbau zu überprüfen.

5.3.4 Signaturen

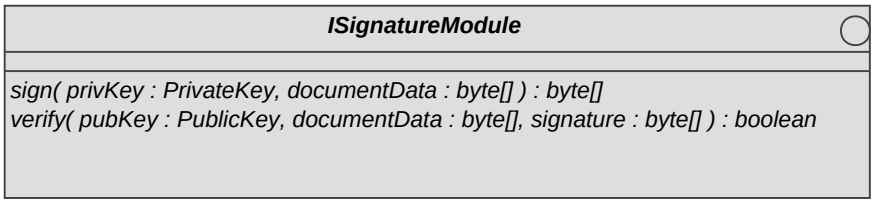

Abbildung 5.2: Klassendiagramm: Signaturmodul

Das Signaturmodul stellt hauptsächlich zwei Methoden zur Verfügung: sign() und verify(). Die Methode sign() dient dazu, eine digitale Signatur für ein angegebenes Dokument zu erstellen. verify() ermöglicht es dagegen, eine bestehende digitale Signatur mit Hilfe des Originaldokuments und des öffentlichen Schlüssels zu prüfen. Während Letzteres nur beim Programmstart zur Anwendung kommt, um die Anwendung auf ihre Konsistenz zu überprüfen, wird die Methode zum Signieren eines Dokuments bei jeder Nutzung eines kritischen Dienstes benötigt.

Sollen die Abrechnungsdaten zur Kontrolle für den Gast auf dem mobilen Endgerät gespeichert werden (siehe Kapitel 5.3.2), so kann dies ebenfalls in diese bzw. in die aufrufende Methode integriert werden.

5.3.5 Dienste

Der Entwurf des Dienstprotokolls orientiert sich am HTTP (HyperText Transfer Protocol), wie es im RFC 1945 spezifiziert ist. Dieses Protokoll dient deshalb als Grundlage, da es sich bereits im Internet für die Übertragung verschiedenster Dienste bewährt hat.

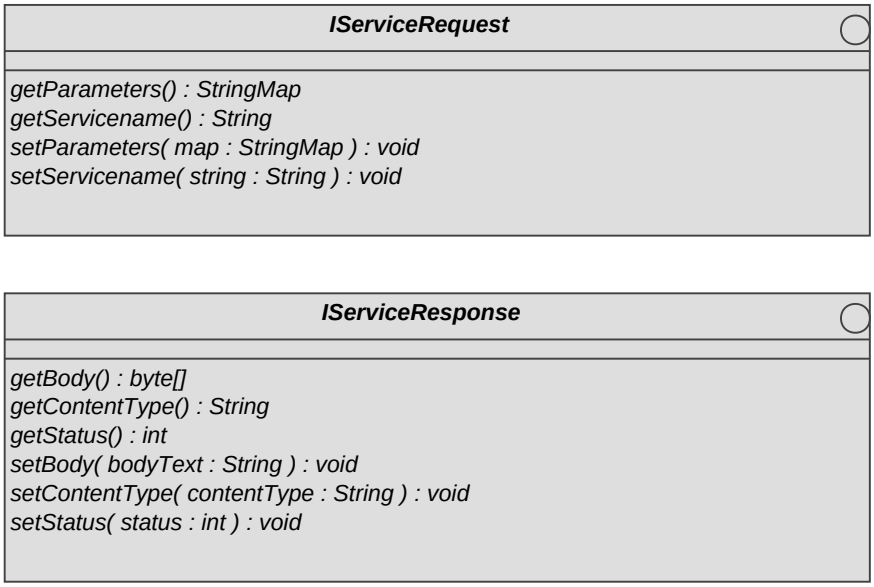

Abbildung 5.3: Klassendiagramm: ServiceRequest und -Response

service request

Der serviceRequest definiert auf abstrakter Ebene, wie eine Dienstanfrage aussehen muss. Demnach enthält eine Dienstanfrage immer den Namen des Dienstes, der in Anspruch genommen werden soll, sowie eine Liste von Parametern, die hier als eine Name-Wert-Zuordnung verstanden wird.

service response

Wie eine Dienstantwort aussehen muss, definiert das Interface IServiceResponse. Eine Antwort besteht aus einem Status, einem Typen und einem so genannten Body.

Der Status gibt an, wie die Nachricht verarbeitet wurde. Er gibt Auskunft darüber, ob und wenn ja, welche Fehler aufgetreten sind, oder ob die Dienstanfrage erfolgreich bearbeitet werden konnte. Folgende Statuscodes sind hier sinnvoll:

1. ClientUpdate - Es sind Softwareupdates oder Patches vorhanden, die eingespielt werden müssen. In diesem Zusammenhang wäre es sinnvoll, wenn der Client, bei jeder Dienstanfrage seine eigene Versionsnummer übertragen würde.
2. ServiceNotFound - Der angeforderte Dienst ist nicht vorhanden.
3. Ok - Der angeforderte Dienst wurde erfolgreich ausgeführt.

4. InternalServerError - Der angeforderte Dienst meldete einen Verarbeitungsfehler.
5. SignatureRequired - Der angeforderte Dienst ist kritisch und benötigt deshalb die digitale Signatur des Benutzers.

Die Option ClientUpdate als Status zu integrieren hat mehrere Gründe. Die Implementierung eines eigenständigen Dienstes hat den Nachteil, dass der Client einen zusätzlichen Kommunikationskanal geöffnet haben muss, der wiederum für Angriffe missbraucht werden könnte. Des Weiteren fordert die Verwaltung eines solchen Kanals zusätzliche Ressourcen, die auf mobilen Endgeräten knapp sind. Nachteil dieses Vorgehens ist jedoch, dass der Server nicht zu jedem beliebigen Zeitpunkt ein Update an das mobile Endgerät übertragen kann, sondern dass der Client erst eine Anfrage senden muss. Diese Anfrage lässt sich jedoch ohne weiteres in den Start der Clientanwendung bzw. in den Verbindungsaufbau zwischen Client und Server integrieren. Es wäre auch denkbar, dass die Anfrage nach Clientupdates als eigener Service implementiert wird. Von dieser Lösung wurde jedoch aus Gründen der Übersichtlichkeit abgesehen. Abschließend sollte noch erwähnt werden, dass es sinnvoll ist, zusammen mit dem Programm auch die dazugehörige Signatur zu übertragen. Dies ist notwendig, wenn eine Signaturprüfung vor der Ausführung von Programmen stattfindet.

Über den Typ kann der Client feststellen, auf welche Art und Weise die im Body übermittelten Daten zu interpretieren sind. Dadurch ist es möglich, nicht nur Parameter und Anzeigedaten an den Client zu übertragen, sondern auch Programme, wie sie beispielsweise für das Update des Clients benötigt werden. Auch die im Kapitel 2 beschriebene Möglichkeit des Spieledownloads sind über diesen Mechanismus möglich.

Der Body enthält, wie bereits beschrieben, Daten, die je nach Typ anders interpretiert werden können.

5.3.6 Kommunikation

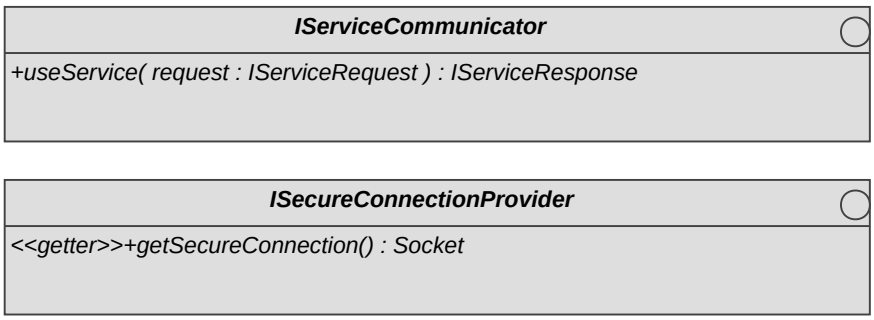

Abbildung 5.4: Klassendiagramm: Kommunikationsklassen

Für die Kommunikation mit dem Server ist nur die Methode useService() notwendig. Diese besorgt sich über den Methodenaufruf getSecureConnection() vom SecureConnectionProvider einen sicheren Kommunikationskanal. Dieser Kanal kann anschließend dazu genutzt werden, die einzelnen Dienstanfragen an den Server zu senden und dessen Antworten zu empfangen. Die Trennung zwischen Übertragung von Daten und Bereitstellung der Verbindung ist sinnvoll, da es dadurch möglich ist das genutzte Authentifizierungs- und Verschlüsselungsverfahren für die Verbindung zu ändern, ohne dass die eigentliche Datenübertragung angepasst werden muss.

5.3.7 Ablauf

Das Zusammenspiel der einzelnen Komponenten soll nun an einigen Diagrammen verdeutlicht werden.

Bevor das Programm genutzt werden kann, erfolgt erst eine Prüfung der eigenen Konsistenz anhand der Anwendungssignatur des PDA-Betreuers (siehe Abbildung 5.5).

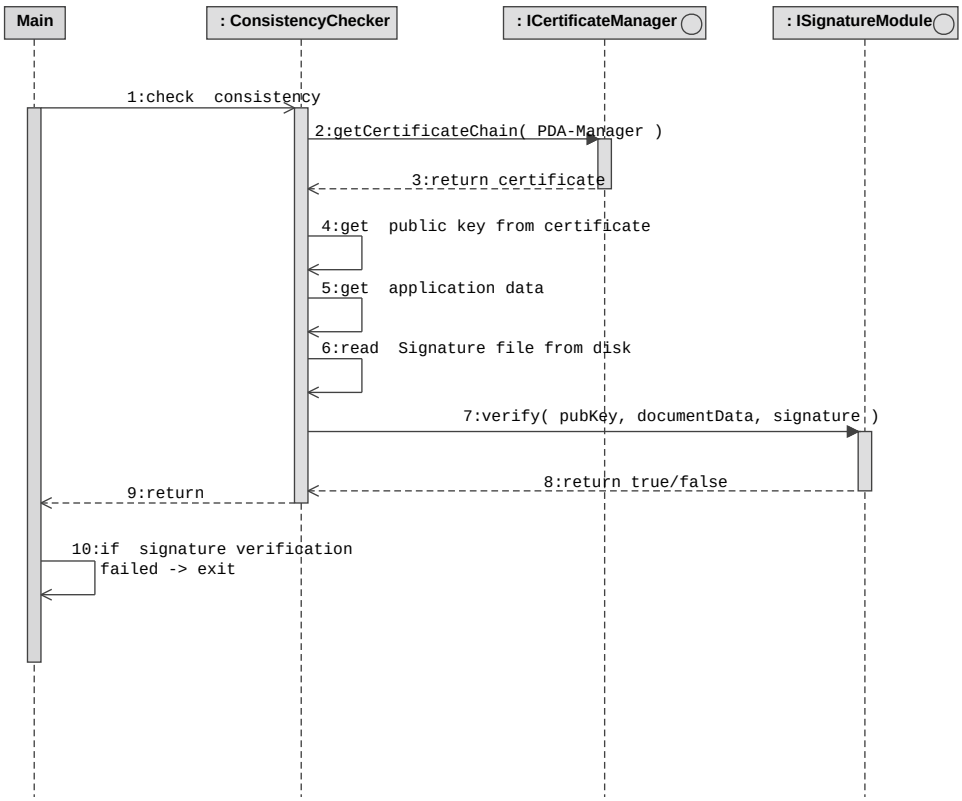

Abbildung 5.5: Sequenzdiagramm: Konsistenz der Anwendung prüfen

Nach einem erfolgreichen Start befindet sich der Benutzer im Hauptmenü. Von dort aus hat er dann die Möglichkeit, einzelne lokale und netzwerkbasierte Dienste zu nutzen. Für letztere muss eine verschlüsselte Netzwerkverbindung aufgebaut werden. Dies kann je nach gewähltem Verfahren anders aussehen. Den groben Ablauf einer zertifikatsbasierten Technologie zeigt Abbildung 5.6.

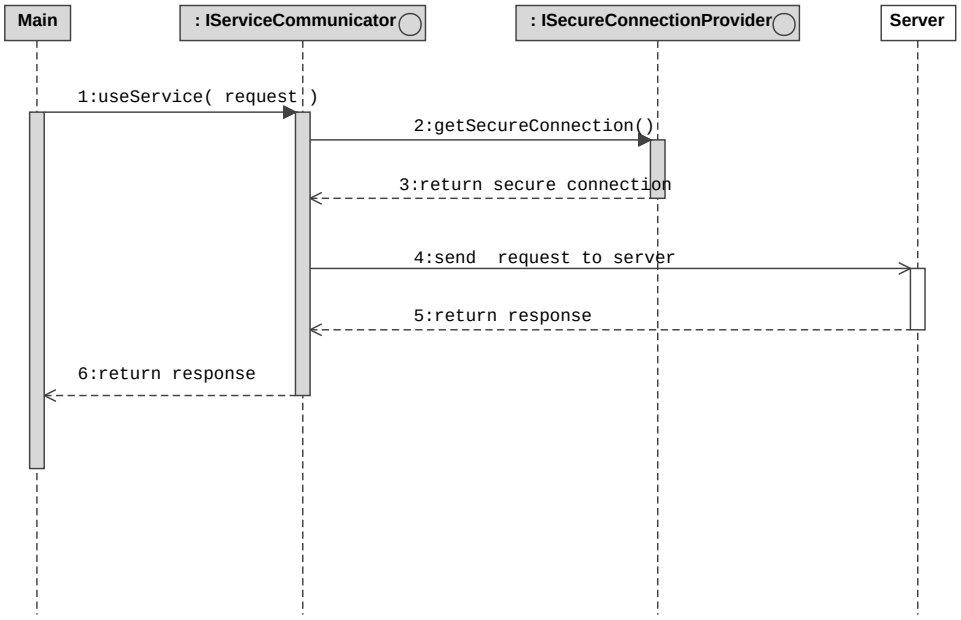

Abbildung 5.6: Sequenzdiagramm: Dienst nutzen

Sollte es sich bei dem angeforderten Dienst um einen kritischen Dienst handeln, so muss diese Anfrage vom Client signiert werden. Wie dies geschehen kann, zeigt Abbildung 5.7.

5.3.8 Server

Auf die Funktionalitäten des Servers soll hier nur kurz eingegangen werden, da diese nicht Schwerpunkt dieser Arbeit sind. Nachfolgend werden einzelne Sicherheitsschichten beschrieben, die serverseitig umgesetzt werden können. Abbildung 5.8 veranschaulicht den gewählten Ansatz.

Auf unterster Ebene kann eine Überprüfung der Netzwerk- bzw. IP-Adresse des PDAs vorgenommen werden. Ob diese Maßnahme sinnvoll ist, hängt vom einzelnen Anwendungsgebiet ab. Es sollte überlegt werden, ob die dadurch erreichte Sicherheit den Aufwand der Realisierung rechtfertigt, da sich sowohl Netzwerkadresse, als auch IP-Adresse relativ leicht vortäuschen lassen.

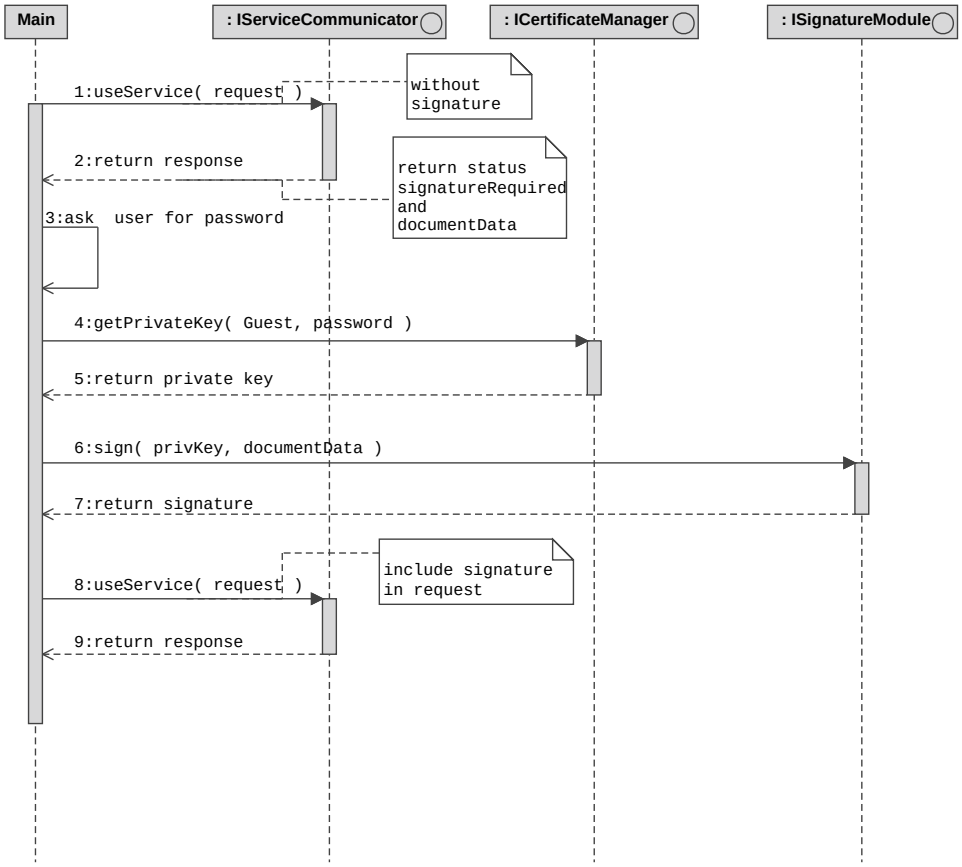

Abbildung 5.7: Sequenzdiagramm: Kritischen Dienst nutzen

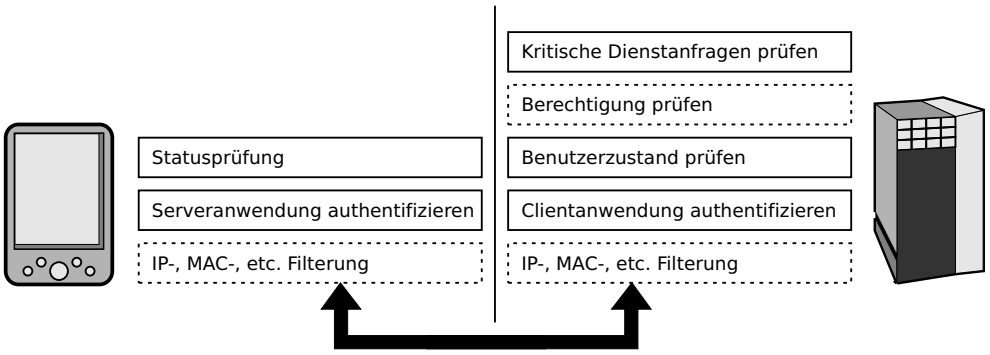

Abbildung 5.8: Sicherheitsschichten auf Client- und Serverseite

Als nächstes sollte die geforderte Clientauthentifizierung über das entsprechende Zertifikat vorgenommen werden.

Eine Ebene höher sollte sofort geprüft werden, ob der zum Client zugehörige Benutzer deaktiviert wurde. Sollte dies der Fall sein, so kann die Anfrage abgebrochen, der Benutzer über diesen Zustand informiert oder aber das mobile Endgerät über die besprochene Möglichkeit des Clientupdates gelöscht werden.

Ist der Benutzer nicht deaktiviert, so sollte als nächstes geprüft werden, ob der Benutzer die notwendigen Rechte zur Nutzung des angeforderten Dienstes besitzt. Im gewählten Szenario ist eine solche Prüfung überflüssig, da es nur gleichberechtigte Gäste gibt; es ist jedoch denkbar, dass dies nicht der Regelfall ist. Um eine gezielte Zugriffskontrolle zu ermöglichen, bietet sich das rollenbasierte Zugriffsmodell [13] an, wie es vom NIST vorgeschlagen wird (NIST, 2003). Dieses Modell basiert auf dem 1996 von R. Sandhu (Sandhu u. a., 1996) vorgeschlagenem Modell und eignet sich besonders gut für den Einsatz in verteilten Systemen (Eckert, 2001a).

Nachdem festgestellt wurde, dass der Benutzer das Recht besitzt, den angeforderten Dienst in Anspruch zu nehmen, sollte das System prüfen, ob es sich um einen kritischen Dienst handelt oder nicht. Sollte es sich um einen kritischen Dienst handeln, so muss geprüft werden, ob mit der Anfrage eine gültige Signatur geschickt wurde. Dazu muss der Server das an den Client gesendete Dokument zwischenspeichern. Sollte die Signatur nicht korrekt sein, so sollte dem Benutzer das zu unterschreibende Dokument erneut vorgelegt oder der Vorgang abgebrochen werden.

Die Überprüfung, ob der Client upgedated werden muss, kann je nach Anwendungsgebiet und Wunsch an beliebiger Stelle eingefügt werden. So kann es sinnvoll sein, den Client upzudaten, auch wenn der Benutzer noch nicht authentifiziert wurde.

5.3.9 Zusammenfassung

Der Entwurf zeigt, wie sich die Anforderungen aus Kapitel 3 zusammen mit den erarbeiteten Maßnahmen aus diesem Kapitel innerhalb einer Clientanwendung realisieren lassen. Aus dem Maßnahmenkatalog wurden dabei die Dienst-, Server- und Cientauthentifizierung, sowie die Überprüfung der Programmsignatur bei Anwendungsstart berücksichtigt. Des Weiteren wurde der Aufbau einer sicheren Kanalverschlüsselung, die Erstellung von digitalen Signaturen für kritische Dienste und die Möglichkeit zum Einspielen von Remoteupdates in den Entwurf mit einbezogen.

Die Aufteilung der Anwendung in einzelne Module erfüllt die Ansprüche nach Modularität, Wiederverwendbarkeit und Erweiterbarkeit. Die Verwendung von X509 V3 Zertifikaten hat zur Folge, dass sich das System auch an zukünftige Ansprüche anpassen lässt. Zudem wir

[13] RBAC (role-based access control)

die Anwendung portabler, da viele Systeme das Speichern und Auslesen solcher Zertifikate unterstützen. Mit dieser Lösung ließen sich sogar bestehende Benutzerzertifikate aus Trust Centern in das System integrieren. Die Abstraktion der Dienstschnittstelle auf Name und Parameter, sowie die Abstraktion der Antworten auf Anfragen auf Typ, Status und Daten lässt ein breites Anwendungsgebiet zu. Das Dienstangebot lässt sich somit flexibel erweitern. Die Verlagerung der Funktionalitäten und Anzeigedaten für Netzwerkdienste auf den Server hat außerdem den Vorteil, dass das Dienstespektrum dynamisch erweitert und verändert werden kann, ohne dass dafür die Clientsoftware angepasst werden muss. Das vereinfacht die Wartung ungemein. Die Verwendung digitaler Signaturen bietet zudem dem Ferienclub eine gesetzliche Grundlage für die Unabstreitbarkeit der Nutzung von Diensten. Der Entwurf berücksichtigt außerdem die Forderung, dass möglichst wenig sensible Daten auf dem mobilen Endgerät gespeichert werden sollten.

Daß sich dieser Entwurf praktisch umsetzen lässt, soll im nachfolgenden Kapitel gezeigt werden.

6 Implementierung

Als erstes soll in diesem Kapitel die Laborumgebung beschrieben werden. Daraufhin wird gezeigt, wie der in Kapitel 5.3 vorgestellte Softwareentwurf innerhalb dieser Laborumgebung prototypisch implementiert wurde. Der Fokus liegt auch hier auf der Client- und nicht auf der Serveranwendung.

6.1 Laborumgebung

6.1.1 PDA - Pocket PC

Als PDA kommt ein HP (Hewlett-Packard)[1] iPAQ Pocket PC H5550 zum Einsatz. Dieser unterstützt sowohl Bluetooth als auch WLAN (Wireless Local Area Network) nach 802.11b, um sich mit Funknetzen zu verbinden. In dem gewählten Szenario dürfte vor allem die angebotene WLAN-Funktionalität von Nutzen sein. Als Betriebssystem kommt Microsoft Windows Mobile 2003 Premium Software for Pocket PC[2] zum Einsatz. Es handelt sich damit um ein typisches Einbenutzerbetriebssystem, welches außerdem Multitasking unterstützt. Alle weiteren technischen Eigenschaften dieses Gerätes stehen auf der Internetseite von Compaq[3] zur Verfügung.

Sicherheitsfeatures

In diesem Abschnitt soll kurz beschrieben werden, welche der geforderten Sicherheitsmaßnahmen aus Kapitel 5.2 der vorgestellte PDA von Haus aus unterstützt.

Zur lokalen Authentifizierung werden neben der üblichen Authentifizierung über Passwort und PIN auch eine Authentifizierung über einen integrierten Fingerabdruckleser angeboten. Sowohl über die FAR des biometrischen Verfahrens, als auch über die Qualität der anderen Verfahren ließen sich keine Informationen auffinden. Es soll jedoch nicht unerwähnt bleiben, dass sich die lokale Authentifizierung vom Benutzer jederzeit ändern und auch

[1] http://www.hp.com
[2] http://www.microsoft.com/windowsmobile/products/pocketpc/default.mspx
[3] http://www.compaq.com

komplett abschalten lässt. Bei einem Hard Reset wird der geräteinterne Speicher komplett mit einem Herstellerimage überschrieben und erfüllt somit die im Entwurf geforderten Eigenschaften. Eingebundene externe Speicherkarten werden jedoch nicht gelöscht. Über einen integrierten Backupmechanismus bietet sich dem PDA-Betreuer die Möglichkeit, auf einfache Weise vorgefertigte Images auf den PDA aufzuspielen. Der PDA-Manager hat die Möglichkeit einen Zeitwert festzulegen, nach dem das Gerät sich selbst ausschaltet, falls in der Zwischenzeit keine Benutzereingaben erfolgt sind. Auch diese Einstellung lässt sich jedoch vom Benutzer ohne weiteres verändern. Virenscanner und Firewalls sind nicht in der Konfiguration enthalten. Für Authentifizierungen im Netzwerk steht ein Zertifikatsspeicher bereit. Über diesen kann z.B. der auf dem Gerät vorinstallierte Browser eine Client- und Serverauthentifizierung für SSL-Verbindungen durchführen und damit gleichzeitig einen verschlüsselten Kommunikationskanal herstellen. Auch der Aufbau eines VPNs über PPTP oder IPSec/L2TP (Layer Two Tunneling Protocol) wird vom System unterstützt. Das Löschen einzelner Programme ist nur eingeschränkt möglich. So lässt sich zwar die Browseranwendung löschen, die Funktionalität zum Kopieren von Anwendungen ist jedoch nicht ohne weiteres entfernbar. Über den vorhandenen SD (Secure Digital)[4] Slot können sowohl externe SD-, SDIO- (Secure Digital Input/Output)[5], und MMC-(MultiMediaCard)[6] Karten benutzt werden. Die Verwendung dieser Karten lässt sich weder verbieten, noch einschränken. Schlimmer noch, es wird die Möglichkeit angeboten, dass Programme auf der Speicherkarte hinterlegt werden, die bei der Einführung der Karte automatisch gestartet werden. Die Ausführung von Programmen lässt sich ebenso wenig einschränken, wie die bereits erwähnte Änderung von kritischen Betriebssystemeinstellungen. Möglichkeiten zum Aktualisieren des PDAs aus der Ferne werden nicht angeboten. Auch Funktionalitäten zur Erstellung von digitalen Signaturen werden nicht angeboten.

6.1.2 Server

Als Server stand ein normaler Desktop PC zur Verfügung. Als Betriebssystem kam Windows XP Professional[7] zum Einsatz. Da die Serversoftware allerdings in Java[8] geschrieben wurde, sollte es kein Problem sein, die Anwendung auch auf anderen Betriebssystemen laufen zu lassen.

6.1.3 Access Point

Bei dem verwendeten AP handelt es sich um ein handelsübliches Gerät, welches neben der 802.11b Unterstützung für WLAN eine Firewall und andere Funktionalitäten anbot. Für die

[4] http://www.sdcard.org
[5] http://www.sdcard.org
[6] http://www.mmca.org
[7] http://www.microsoft.com/windowsxp/default.asp
[8] http://java.sun.com

prototypische Implementierung wurde jedoch auf den Einsatz dieser weiteren Funktionen verzichtet. Die Verbindung zwischen Access Point und Server wurde über eine 100 MBit Ethernet Leitung hergestellt.

6.2 Implementierung des Softwareentwurfs

Als Programmiersprache für die Clientsoftware wurde Java von Sun Microsystems[9] gewählt, da diese insbesondere die Ansprüche nach Portabilität erfüllt. Als Virtuelle Maschine kam die dem PDA beiliegende, von der Firma Insignia Solutions[10] entwickelte Software JeodeRuntime™ zum Einsatz. Der Client erfüllt damit die PersonalJava 1.2 Spezifikation von Sun Microsystems.

Für den Prototypen wurden neben der Anwendungslogik für die Benutzerschnittstelle alle im Entwurf beschriebenen Interfaces implementiert. Außerdem wurden einige Beispieldienste entworfen und realisiert, um die Funktionsweise der Clientanwendung zu veranschaulichen. Als lokaler Dienst wurde ein Reiseführer erstellt, welcher neben Text auch Grafiken darstellt. Des Weiteren wurden Netzwerkdienste implementiert, welche es dem Benutzer erlauben, sich über stattfindende Konzerte zu informieren. Das Buchen von Tickets für diese Konzerte wurde als kritischer Dienst umgesetzt. Um die Übertragung von Programmen zu demonstrieren, wurde ein weiterer Dienst implementiert, welcher immer den Status „ClientUpdate“ sowie eine Beispielanwendung als Body an den Client zurückliefert.

6.2.1 Verteilung der Daten

Da nach dem Entwurf die Verwaltung und Überprüfung der Abrechnungs- und Zustandsdaten serverseitig vorgenommen werden sollte, wurden diese im Prototypen nicht implementiert.

PDA-Anwendungsdaten

Die Übertragung der Anzeigedaten erfolgt mit Hilfe von so genannten Thinlets. Dies ist ein XML-basiertes Format zur Darstellung von graphischen Benutzeroberflächen. Wie eine solche Definition aussehen kann, zeigt Listing 6.1.

Listing 6.1: Beispiel Thinlet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<panel property="encoding=ISO-8859-1;buildpath=;cls=" columns="1">
```

[9] http://www.sun.com
[10] http://www.insignia.com/

```
  <label halign="center" text="Palm Club" />
  <panel weightx="1" >
    <button
        name="localWelcome"
        weightx="1"
        icon="/icons/Home16.gif"
        action="useService(localWelcome.name,thinlet)"
    />
    <button
        name="exit"
        weightx="1"
        icon="/icons/Stop16.gif"
        action="useService(exit.name,thinlet)"
    />
    <button
        name="localPreferences"
        weightx="1"
        icon="/icons/Preferences16.gif"
        action="useService(localPreferences.name,thinlet)"
    />
  </panel>
</panel>
```

Abbildung 6.1 zeigt, wie die XML-Anzeigedaten durch den Thinletparser interpretiert und angezeigt werden. Zusätzlich zum XML-Dokument kann ein EventHandler an den Parser übergeben werden. An diesen Handler werden daraufhin alle vom Benutzer getätigten Aktionen weitergeleitet.

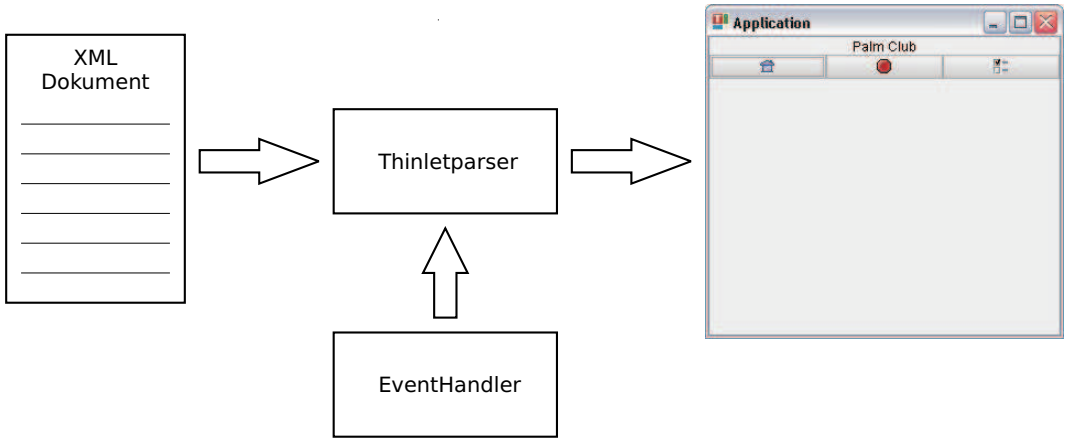

Abbildung 6.1: Thinletarchitektur

Während der Implementierung hat sich gezeigt, dass das die Verwendung der Bibliothek für dynamische Dienste einige Nachteile hat. So ist es nicht ohne weiteres möglich, die Daten aller angezeigten Eingabefelder abzufragen, um daraus dynamisch eine Parameterliste aufzubauen. Auch das Übertragen und Anzeigen von Grafiken gestaltet sich schwierig.

Für die Erstellung von Diensten, die lediglich Daten anzeigen und kaum Benutzereingaben erfordern, ist diese Bibliothek jedoch bestens geeignet.

6.2.2 Zertifikatsverwaltung

Da die PersonalJava Spezifikation nur sehr wenige kryptographische Funktionen zur Verfügung stellt, wurde für diesen Zweck eine externe Bibliothek eingebunden. Zum Einsatz kam die Java Cryptography Extension Micro Edition von IAIK (institute for applied information processing and communications)[11] in der Version 3.0. Diese Bibliothek bietet zudem die Möglichkeiten, Zertifikate und private Schlüssel in einem dafür vorgesehenen passwortgeschützten Keystore zu speichern. Von dieser Möglichkeit wurde bei der Implementierung Gebrauch gemacht.

Für den Prototypen wurden die bereits angesprochenen X509 Zertifikate verwendet, wobei als Algorithmus zur Schlüsselgenerierung MD5 (Message Digest Algorithm 5) mit RSA[12] zum Einsatz kam.

6.2.3 Signaturen

Zur Erstellung und Überprüfung von Signaturen wurden ebenfalls die von der Java Cryptography Extension Micro Edition angebotenen Funktionen verwendet. Die für die Bildung der Signaturen erforderlichen Hashwerte wurden über MD5 erstellt.

Bei kritischen Dienstanfragen wird das zu signierende Dokument dem Benutzer zusammen mit einer Passwortabfrage präsentiert. Das eingegebene Passwort wird daraufhin genutzt, um auf den entsprechenden Keystore zuzugreifen und den privaten Schlüssel des Benutzers zu lesen. Als Dokument wird im Prototypen lediglich ein String übertragen, welcher alle Parameter der Dienstanfrage enthält (siehe Abbildung 6.2).

Die eigene Signaturprüfung der Anwendung wurde ebenfalls implementiert. Dazu steht ein Tool für den PDA-Betreuer bereit, mit dem er Signaturdateien für beliebige Anwendungen erstellen kann. Diese Signaturdatei muß daraufhin mit der Anwendung zusammen auf das mobile Endgerät übertragen werden (siehe Abbildung 6.3).

[11] graz university of technology - http://www.iaik.tugraz.at/

[12] RSA ist ein 1977 entwickeltes asymmetrisches Verschlüsselungsverfahren, welches nach seinen Erfindern Ronald L. Rivest, Adi Shamir und Leonard Adleman benannt wurde.

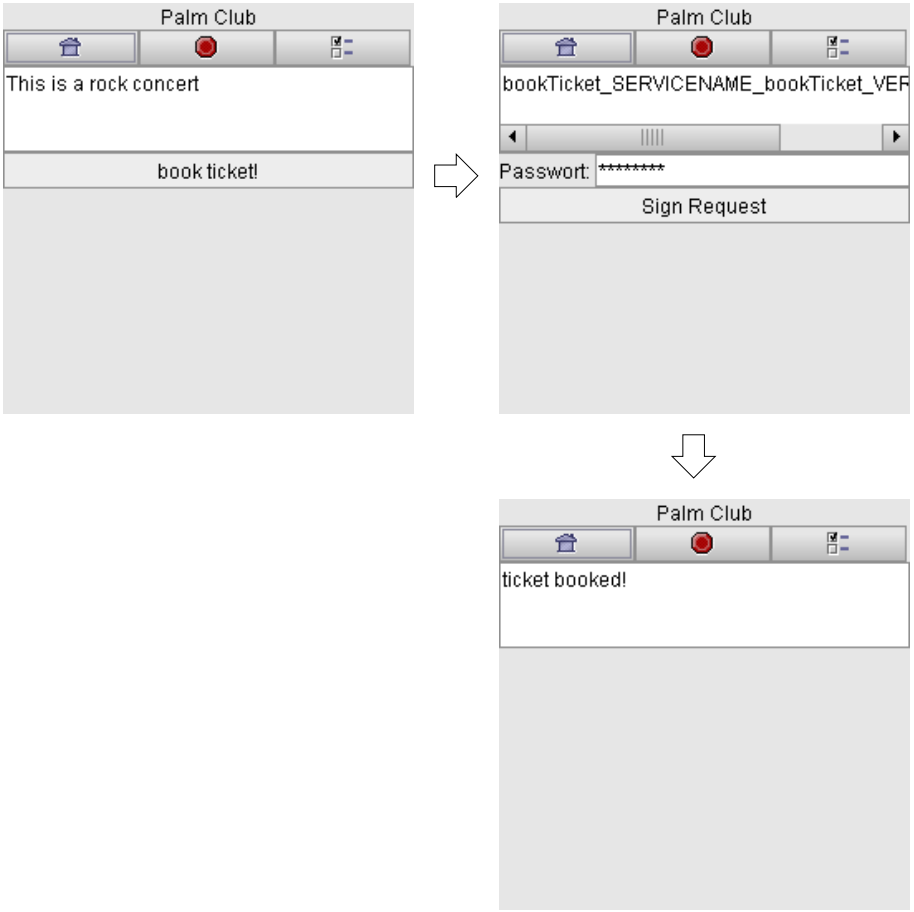

Abbildung 6.2: Screenshot: kritischen Dienst nutzen

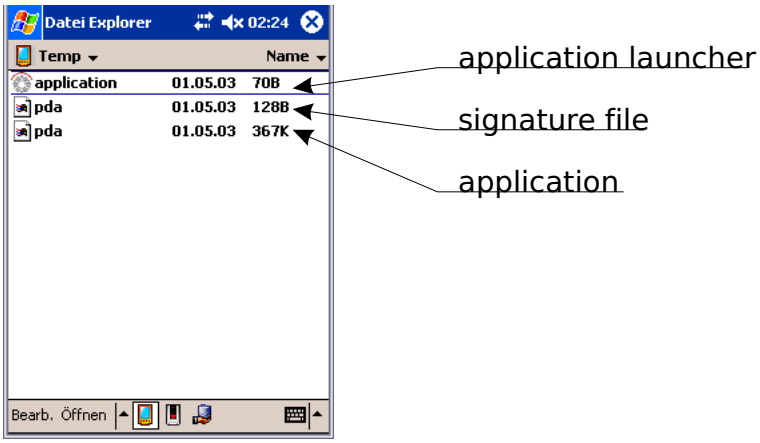

Abbildung 6.3: Screenshot: Dateisystem mit Signaturdatei

6.2.4 Dienste

Die Implementierung der Dienstanfragen und -antworten orientiert sich wie der Entwurf am HTTP-Standard. Als Status Codes wurden demnach einige der HTTP-Stati übernommen. Dazu zählen unter anderem 404 für nicht gefundene Dienste und 500 für interne Server Fehler. Neu hinzugefügt wurden die Stati 450 für das Fehlen einer gültigen Signatur und 451 für die Übertragung eines Clientupdates.

Als Typen wurden Strings verwendet, die den bekannten MIME- (Multipurpose Internet Mail Extensions) Typen ähneln.

Abbildung 6.4 zeigt Bildschirmfotos einiger angebotenen Beispieldienste.

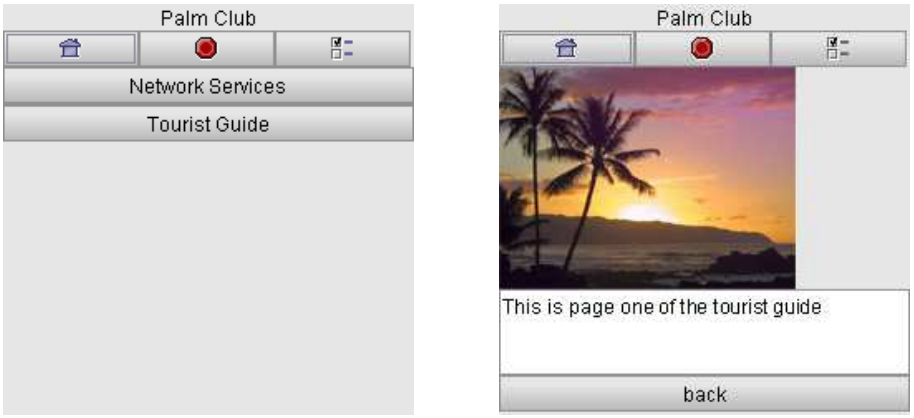

Abbildung 6.4: Screenshot: Allgemeine Dienste

Um das Clientupdate zu prüfen, wurde die Übertragung einer Beispielanwendung implementiert. Der Test ist erfolgreich verlaufen, so dass davon ausgegangen werden kann, dass dies auch für verschiedene andere Programme funktioniert. Das einzige Problem, welches in diesem Zusammenhang auftrat, war, dass die übertragene Anwendung nicht direkt aus der Clientanwendung heraus ausgeführt werden konnte. Aus diesem Grund wird der Benutzer der Clientanwendung nach der Übertragung des Programms darauf hingewiesen, dass er das übertragene Programm manuell starten muss (siehe Abbildung 6.5). Außerdem erfordert die Übertragung selbst kleiner Anwendung eine relativ lange Wartezeit.

6.2.5 Kommunikation

Um eine Client- und Serverauthentifizierung durchzuführen, sowie um einen verschlüsselten Kommunikationskanal zwischen Client und Server herzustellen, wurde im Prototypen SSL mit aktivierter Client- und Serverauthentifizierung verwendet. Dazu war die Einbindung einer weiteren externen Bibliothek notwendig. Zum Einsatz kam iSaSiLk Micro Edition von

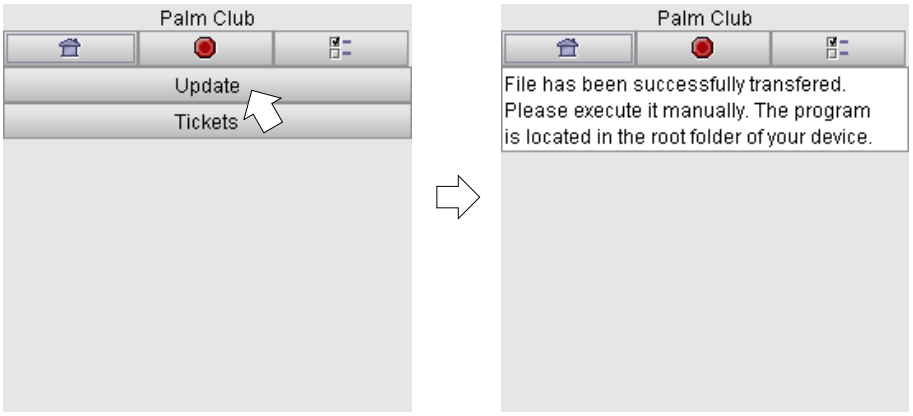

Abbildung 6.5: Screenshot: Erfolgreiche Übertragung einer Anwendung

IAIK in der Version 3.0. Es sollte jedoch ohne weiteres möglich sein, durch das Austauschen der ConnectionProvider Klasse auf andere Verfahren wie beispielsweise Kerberos zu wechseln.

Die Übertragung selbst erfolgt über die von Java bereitgestellten DataOutput- bzw. DataInputStreams.

6.2.6 Server

Die Serveranwendung wurde nur sehr rudimentär implementiert, da sie nicht Gegenstand der Arbeit war. Sie wurde ebenfalls in Java implementiert und bildete nur die Grundfunktionalitäten ab, die für den Betrieb der Clientsoftware nötig waren.

6.2.7 Zusammenfassung

Die Implementierung hat gezeigt, dass sich der Entwurf relativ problemlos in die Praxis umsetzen lässt. Die kurze Untersuchung des eingesetzten PDAs im Abschnitt 6.1 hat jedoch auch gezeigt, dass die heutigen Betriebssysteme noch einige Wünsche im Bereich der Sicherheit offen lassen.

Die Wahl der Programmiersprache Java hat sich zudem als überaus sinnvoll erwiesen, da sich dadurch die Software fast komplett auf einem normalen Desktop PC implementieren und testen ließ. Die anschließende Übertragung auf den Pocket PC verlief weitestgehend reibungslos. Die Anfangs geforderte Portabilität ist somit realisiert worden.

Die erwartete Wiederverwendbarkeit einzelner Komponenten ließ sich nicht vollständig realisieren, da sich die APIs (Application Programming Interfaces) der verwendeten Kryptographie-Bibliotheken auf Server- und Clientseite geringfügig unterschieden. Auch die clientseitige Verarbeitungsgeschwindigkeit einzelner Anwendungsfälle ließ zu wünschen übrig. Des Weiteren bereitete die serverseitige Kryptographie-Bibliothek Schwierigkeiten beim Auslesen der Clientzertifikate. Die beschriebenen Probleme haben jedoch nichts mit dem Entwurf an sich zu tun.

Vorteilhaft hat sich die Orientierung am HTTP-Standard erwiesen, da es damit sehr einfach ist, neue Netzwerkdienste hinzuzufügen, ohne dass die Clientanwendung verändert werden muß. Durch den Einsatz von Thinlets war es außerdem auf einfache Art und Weise möglich, lokale Dienste zu verändern und zu erweitern, ohne dass dafür in den Quelltext der Software eingegriffen werden mußte. Die Speicherung der Anzeigedaten in XML hat des Weiteren zu einer sauberen Trennung zwischen Präsentation und Logik geführt. Den Ansprüchen nach Wartbarkeit und Erweiterbarkeit wird die Implementierung demnach vollends gerecht.

Zusammenfassend läßt sich feststellen, dass sich der Entwurf bewährt hat. Es hat sich außerdem gezeigt, dass sich der Entwurf bereits mit heutigen Mitteln ohne weiteres in die Praxis umsetzen läßt.

7 Zusammenfassung und Ausblick

7.1 Zusammenfassung

Ziel dieser Arbeit war der Entwurf einer Sicherheitsarchitektur für den Einsatz mobiler Endgeräte. Dieses Ziel wurde mit der vorliegenden Arbeit erreicht. Dazu wurde zuerst ein Szenario festgelegt, anhand dessen die Analyse stattfand. In der Analyse wurden als erstes die Komponenten und IT-Anwendungen mit ihren dazugehörigen Informationen herausgearbeitet. Anschließend wurden die Daten hinsichtlich ihres Schutzbedürfnisses kategorisiert. In einer Bedrohungsanalyse wurden daraufhin systematisch alle Bedrohungen des Systems zusammengetragen und detailiert aufgeschlüsselt. Im Entwurf konnten somit für alle Bedrohungen Maßnahmen definiert werden, um das System abzusichern. Anschließend wurden die erarbeiteten Maßnahmen in einen einheitlichen Maßnahmenkatalog überführt. Dann wurde eine Clientsoftware entworfen, welche die wichtigsten clientseitigen Maßnahmen der Anwendungsebene berücksichtigt und die sichere Nutzung von Diensten gewährleistet. Im darauf folgenden Kapitel wurde die konzipierte Clientanwendung inklusive einiger Beispieldienste implementiert. Dieser Schritt hat nachhaltig gezeigt, dass sich die ausgewählten Maßnahmen bereits mit dem heutigen Stand der Technik realisieren lassen.

7.2 Fazit

Diese Arbeit hat gezeigt, dass sich alle wesentlichen Bedrohungen des gewählten Szenarios durch geeignete Maßnahmen abwenden lassen. Die Maßnahmen betreffen dabei sowohl die Anwendungs- als auch die Betriebssystemebene. Während der realisierte Prototyp gezeigt hat, dass sich die Maßnahmen auf Anwendungsebene bereits heute problemlos umsetzen lassen, hat eine kurze Untersuchung des bereitgestellten PDAs gezeigt, dass das Pocket PC Betriebssystem für mobile Endgeräte dem Thema Sicherheit nicht genügend Aufmerksamkeit schenkt. Solange die Betriebssysteme keine grundsätzlichen Möglichkeiten bieten, Sicherheitsrichtlinien für einzelne Benutzer zu setzen, sind eine Reihe der notwendigen Maßnahmen nicht ohne weiteres realisierbar. Interessante Ansätze

auf Betriebssystemebene wurden bereits entworfen und prototypisch realisiert[1][2]. Es bleibt abzuwarten, in wie weit sich solche Ansätze durchsetzen. Für eine Sicherheitsarchitektur ist es jedoch zwingend notwendig, dass alle relevanten Maßnahmen umgesetzt werden, da die Sicherheit eines Systems bekanntlich immer nur so gut ist, wie ihr schwächstes Glied.

7.3 Ausblick

Ziel dieses Kapitels ist es zu zeigen, wie sich die angestellten Untersuchungen fortführen ließen.

Da sich im Verlauf der Arbeit gezeigt hat, dass Einbenutzerbetriebssysteme nur durch eine ganze Reihe von Maßnahmen abgesichert werden können, wäre es interessant, zu untersuchen, wie die bereits für PDAs und Web-Tabletts verfügbaren Linux Versionen den Aufwand für die Erstellung eines sicheren mobilen Endgerätes verringern können. In diesem Zusammenhang sei auf die Studienarbeit von Lars Mählmann[3] hingewiesen.

Auf dem Gebiet der Clientprogrammierung wäre außerdem eine Untersuchung der Sicherheitsfunktionen der von Sun Microsystems entwickelten MIDP- (Mobile Information Device Profile) Spezifikation denkbar. Diese Spezifikation wurde extra für den Einsatz von Java auf mobilen Endgeräten entwickelt und stellt den Nachfolger der in dieser Arbeit eingesetzten PersonalJava Spezifikation dar.

Die Erweiterung des Szenarios um weitere Rollen wie z.B. Angestellte, die über ein mobiles Endgerät besondere Dienste in Anspruch nehmen können, ist ebenfalls untersuchenswert. So wäre es z.B. möglich, die Dienstnutzung um ein rollenbasiertes Zugriffsverfahren zu erweitern. Auch das automatische Sperren des mobilen Endgerätes außerhalb der Arbeitszeiten eines Angestellten wäre denkbar.

Es könnte außerdem über die Einführung von Diensten nachgedacht werden, die vom Benutzer erst an der Rezeption freigeschaltet werden müssen. Genauso wäre es denkbar, dass bestimmte Dienste explizit vom Benutzer gesperrt werden können, falls dieser das mobile Endgerät z.B. an seine Kinder weitergeben möchte.

Serverseitig wäre zudem interessant, wie der Ferienclub auf eine sichere Art und Weise die Dienste von Drittanbietern in das Szenario integrieren kann und wie sich die Abrechnung zwischen beiden realisieren lässt.

---

[1] Das LucaOS-Projekt des universitätsübergreifenden Schwerpunktprogramms Sicherheit in der Informations- und Kommunikationstechnik (SPP Sicherheit - `http://www13.in.tum.de/SPP/index.shtml`) beschäftigt sich mit der Konzeption eines sicheren Betriebssystems für mobile Endgeräte.

[2] Ein auf der International Conference on Security and Management 2003 vorgestelltes Papier stellt einen Lösungsansatz für die Verwaltung von Sicherheitsrichtlinien auf Handheld Geräten vor (Jansen u. a., 2003).

[3] Lars Mählmann schreibt in diesem Zusammenhang eine Studienarbeit an der Hochschule für Angewandte Wissenschaften Hamburg.

Im Bereich der Authentifizierung könnte zudem untersucht werden, wie sich Chipkarten oder RFIDs (Radio Frequency Identification) in das Szenario einbinden ließen. Die Integration von RFIDs in Armbändern hätte den Vorteil, dass bereits viele Urlauber an die Verwendung von Armbändern zur Erkennung der Clubzugehörigkeit gewöhnt sind. Chipkarten hätten dagegen den Vorteil, dass viele Benutzer wahrscheinlich den Wert einer solchen Karte durch die Erfahrungen mit EC-Karten besser einschätzen könnten.

Andere Szenarios könnten kritische Dienste erfordern, die sich komplett lokal auf dem mobilen Endgerät befinden. In wie weit sich dies überhaupt auf Einbenutzerbetriebssystemen realisieren ließe, bietet ebenfalls Raum für weitere Untersuchungen.

Die Übertragung des Entwurfs in andere, weitaus kritischere Szenarios, wie beispielsweise Krankenhäuser, sollte ebenfalls weiter betrachtet werden. Diesbezüglich sei auf die Arbeit von Roman Bartnik[4] verwiesen.

Glaubt man den überall publizierten Ankündigungen, so werden in Zukunft immer mehr geschäftliche Transaktionen über mobile Endgeräte abgewickelt werden. Dies kann jedoch nur erfolgreich stattfinden, wenn die dazu genutzten Geräte und Anwendungen sicher sind. Welche Sicherheitsprobleme in diesem Zusammenhang bewältigt werden müssen und wie sich diese durch geeignete Maßnahmen bereits mit dem heutigen Stand der Technik in den Griff bekommen lassen wurde in dieser Arbeit erläutert.

[4] Roman Bartnik untersucht im Rahmen seiner Studienarbeit an der Hochschule für Angewandte Wissenschaften Hamburg, welche Anforderungen erfüllt sein müssen, um mobile Endgeräte in Krankenhäusern einzusetzen.

A Anhang

A.1 Glossar

Asymmetrische Verschlüsselung

Bei der asymmetrischen Verschlüsselung werden, im Gegensatz zur symmetrischen Verschlüsselung, für die Ver- und Entschlüsselung unterschiedliche Schlüssel verwendet. Die einzelnen Schlüssel werden dabei, je nach Verwendungszweck, entweder als öffentlich oder privat bezeichnet.

Authentifizierung

Die Authentifizierung befasst sich mit der Überprüfung der angegebenen Identität einer Entität.

Brute-Force-Verfahren

Bei einem Brute-Force-Verfahren wird versucht, ein Problem dadurch zu lösen, dass alle potentiellen Lösungen durchprobiert werden.

Digitale Signatur

Die Digitale Signatur beschreibt ein kryptographisches Verfahren, mit dem die Urheberschaft eines Dokumentes verifiziert werden kann.

Einweg-Hashfunktion

Eine Einweg-Hashfunktion unterscheidet sich von einer Hashfunktion dadurch, dass sie gewährleistet, dass es für einen gegebenen Hashwert mit vernünftigem Aufwand nicht möglich ist, eine Bytefolge zu konstruieren, die denselben Hashwert produziert.

Hashfunktion

Eine Funktion, die aus umfangreichen Daten einen eindeutigen, komprimierten Wert berechnet.

Identifikation

Bei der Indentifikation handelt es sich um den Vorgang, mit dem ein System eine Entität erkennt.

Kompromittierung

Ein Vorgang, bei dem eine verschlüsselte Nachricht ohne Kenntnis des Schlüssels offengelegt wird.

Man-in-the-middle-Attacke

Bei einem solchen Angriff schaltet sich ein böswilliger Dritter zwischen die Kommunikation zweier oder mehrerer Netzwerkteilnehmern. Er kann die ausgetauschten Nachrichten nach Belieben manipulieren und einsehen.

Öffentlicher Schlüssel (Public Key)

Asymmetrischer Schlüssel, der für die Verschlüsselung bzw. die Verifizierung einer Digitalen Signatur verwendet wird. Er bildet das Gegenstück zum privaten Schlüssel.

Obfuscator

Nach (Collberg u. a., 1998) handelt es sich dabei um eine Anwendung, welche Programme durch Codetransformationen in äquivalente Programme konvertiert. Die durchgeführten Transformationen sind so gewählt, dass sie Angriffe durch Reverse Engineering erschweren.

Privater Schlüssel (Private Key)

Asymmetrischer Schlüssel, der für die Entschlüsselung bzw. das Signieren von Daten verwendet wird. Er bildet das Gegenstück zum öffentlichen Schlüssel.

Reverse Engineering

Ein Vorgehen, bei dem versucht wird, die Funktionsweise einer Anwendung, von der kein Quelltext zur Verfügung steht, zu analysieren.

RSA

RSA ist ein 1977 entwickeltes asymmetrisches Verschlüsselungsverfahren, welches nach seinen Erfindern Ronald L. Rivest, Adi Shamir und Leonard Adleman benannt wurde.

Zugangskontrollverfahren

Bezeichnung für Verfahren, die sich mit der Identifikation und Authentifizierung von Benutzern eines Systems beschäftigen.

A.2 Inhalt der CD-ROM

Im Wurzelverzeichnis der CD-ROM befinden sich die Verzeichnisse „Diplomarbeit“, „Implementierung“ und „Quellen“.

Das Verzeichnis „Diplomarbeit“ enthält die Diplomarbeit im PDF-Format.

Im Verzeichnis „Implementierung“ befindet sich die prototypische Implementierung der Client- und Serversoftware. Es gliedert sich in die Unterverzeichnisse „Server“, „Client“ und „Common“. Bei den einzelnen Verzeichnissen handelt es sich um Projekte, die mit der Entwicklungsumgebung Eclipse[1] Version 3.0M7 erstellt wurden. Das „Common“-Verzeichnis beinhaltet alle Quelltexte, die sowohl für das „Client“-, als auch für das „Server“-Projekt benötigt werden. Jedes einzelne Verzeichnis enthält neben Projektmetadaten die Quelltexte der Implementierung. Die „docs“-Verzeichnisse enthalten einige wichtige Informationen über die jeweiligen Projekte.

Das Verzeichnis „Quellen“ beinhaltet einige der Dokumente, die im Literaturverzeichnis erwähnt werden.

[1] http://www.eclipse.org

Literaturverzeichnis

[WLANVisual 2002] *Wireless Network Visualization Project*. University of Kansas - Information and Telecommunication Technology Center. 2002. – URL http://www.ittc.ku.edu/wlan. – visited 01.03.2004

[BBCvirus 2003] *Mobile virus threat looms large*. BBC News. January 2003. – URL http://news.bbc.co.uk/1/hi/technology/2690253.stm. – visited 01.03.2004

[Adams und Lloyd 2002] ADAMS, Carlisle ; LLOYD, Steve: *Understanding PKI: Concepts, Standards, and Deployment Considerations, Second Edition*. Addison Wesley, November 2002. – ISBN 0672323915

[Allen 2001] ALLEN, Julia H.: *The CERT Guide to System and Network Security Practices*. Addison Wesley, 2001 (SEI Series). – ISBN 0-201-73723-X

[Bachfeld 2003] BACHFELD, Daniel: *W32.Blaster befällt Hunderttausende von PCs*. heise online. August 2003. – URL http://www.heise.de/security/news/meldung/39377. – visited 15.04.2004

[BSI 2002] BSI: *IT-Grundschutzhandbuch*. Bundesanzeiger Verlag, 2002. – Bundesamt für Sicherheit in der Informationstechnik. – ISBN 3-88784-915-9

[CentralNic 2001] CENTRALNIC: *Password Clues*. http://www.centralnic.com/page.php?pid=73. 2001. – visited 01.02.2004

[Collberg und Thomborson 2002] COLLBERG, Christian S. ; THOMBORSON, Clark: Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection. In: *IEEE Transactions on Software Engineering* Bd. 28, URL http://citeseer.nj.nec.com/collberg02watermarking.html, August 2002, S. 735–746. – visited 01.03.2004

[Collberg u. a. 1998] COLLBERG, Christian S. ; THOMBORSON, Clark D. ; LOW, Douglas: Breaking Abstractions and Unstructuring Data Structures. In: *International Conference on Computer Languages*, URL http://citeseer.ist.psu.edu/collberg98breaking.html, 1998, S. 28–38. – visited 10.03.2004

[David 2002] DAVID: Securing Wireless Communications experience, April 2002. – SANS2002 Technical Conference

[Dedo 2002] DEDO, Douglas: *Security on the Pocket PC*. Mobil Devices Devision - Microsoft Corporation. May 2002. – URL http://www.microsoft.com/windowsmobile/resources/whitepapers/security.mspx. – visited 01.02.2004

[Dedo 2004] DEDO, Douglas: *Windows Mobile-Based Devices and Security: Protecting Sensitive Business Information*. Mobile Devices Division - Microsoft Corporation. February 2004

[Donath 2000] DONATH, Andreas: *Love Letter: Killergrüße eines E-Mail-Wurms*. Golem.de. May 2000. – URL http://www.golem.de/0005/7556.html. – visited 15.04.2004

[Dumke] DUMKE: *UML Tutorial*. http://ivs.cs.uni-magdeburg.de/~dumke/UML/. – visited 01.03.2004

[Eckert 2001a] ECKERT, Claudia: *IT-Sicherheit : Konzepte - Verfahren - Protokolle*. Oldenbourg, 2001. – ISBN 3-486-25298-4

[Eckert 2001b] ECKERT, Claudia: Zur Sicherheit mobiler persönlicher Endgeräte - Eine Bestandsaufnahme. In: *Kommunikationssicherheit im Zeichen des Internet*, March 2001, S. 204–217. – Arbeitskonferenz Kommunikationssicherheit, SAP University Rot

[Geier 2002] GEIER, Jim: *RF Site Survey Steps*. Jupitermedia Corporation. May 2002. – URL http://www.wi-fiplanet.com/tutorials/article.php/1116311. – visited 22.03.2004

[Ghosh und Swaminatha 2001] GHOSH, A. K. ; SWAMINATHA, T. M.: Software Security and Privacy Risks in Mobile E-Commerce. In: *Communications of the ACM* 44 (2001), February, Nr. 2, S. 51–57

[Graff und Wyk 2003] GRAFF, Mark G. ; WYK, Kenneth R. V.: *Secure Coding: Principles and Practices*. O'Reilly & Associates, Inc., July 2003. – ISBN 0596002424

[Howard und LeBlanc 2002] HOWARD, Michael ; LEBLANC, David C.: *Writing Secure Code - Second Edition*. Microsoft Press, December 2002. – ISBN 0735617228

[IZT u. a. 2001] IZT ; SFZ ; IAT: *Entwicklung und zukünftige Bedeutung mobiler Multimediadienste - Werkstattbericht Nr. 49*. December 2001

[Jansen u. a. 2003] JANSEN ; KARYGIANNIS ; IORGA ; GARVRILA ; KOROLEV: Security Policy Management for Handheld Devices. International Conference on Security and Management 2003, June 2003. – visited 01.03.2004

[Jansen 2003] JANSEN, Wayne A.: *Authenticating Users on Handheld Devices*. May 2003. – URL http://csrc.nist.gov/mobilesecurity/Publications/PP-AuthenticatingUsersOnPDAs.pdf. – visited 01.03.2004

[Karygiannis und Owens 2002] KARYGIANNIS, Tom ; OWENS, Les: *Wireless Network Security - 802.11, Bluetooth and Handheld Devices*. November 2002. – URL http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf. – visited 01.03.2004

[Kerckhoff 1883] KERCKHOFF, Auguste: La cryptographie militaire. In: *Journal des sciences militaires* IX (1883)

[Kersten 1991] KERSTEN, Heinrich: *Einführung in die Computersicherheit*. Oldenbourg, 1991 (Sicherheit in der Informationstechnik). – ISBN 3-486-21873-5

[Krause und Tipton 1998] KRAUSE, Micki ; TIPTON, Harold F.: *Handbook of Information Security Management*. CRC Press LLC, January 1998. – ISBN 0849399475

[Kuri 2004] KURI, Jürgen: *Neuer Wurm Novarg/Mydoom verbreitet sich schnell*. heise online. January 2004. – URL http://www.heise.de/security/news/meldung/44035. – visited 15.04.2004

[Kurose und Ross 2001] KUROSE, James ; ROSS, Keith: *Computernetze - Ein Top-Down-Ansatz mit Schwerpunkt Internet*. Addison Wesley, 2001. – ISBN 3-8273-7017-5

[Lyon 2002] LYON: *THE DILEMMA OF PDA SECURITY*. http://www.sans.org/rr/papers/41/257.pdf. 2002. – visited 01.03.2004

[NIST 2003] NIST: *Role Based Access Control, DRAFT*. http://csrc.nist.gov/rbac/. April 2003. – visited 01.03.2004

[Ornaghi und Valleri 2003] ORNAGHI, Alberto ; VALLERI, Marco: Man in the middle attacks Blackhat Conference - USA (Veranst.), 2003

[Peikari und Chuvakin 2004] PEIKARI, Cyrus ; CHUVAKIN, Anton: *Security Warrior*. O'Reilly & Associates, Inc., January 2004. – ISBN 0-596-00545-8

[PocketPCCentral] POCKETPCCENTRAL: *The iPAQ Secure Digital Slot*. – URL http://www.pocketpccentral.net/help/ipaqsdcard.htm. – visited 12.03.2004

[Price 2003] PRICE, Richard: *The PDA as a Threat Vector*. SANS Institute. March 2003. – URL http://www.sans.org/rr/papers/41/998.pdf. – visited 22.01.2004

[Russel und Gangemi 1991] RUSSEL, Deborah ; GANGEMI, G.T.: *Computer Security Basics*. O'Reilly & Associates, Inc., 1991 (Computer Security). – ISBN 0-937175-71-4

[Russell u. a. 2003] RUSSELL, Ryan ; MULLEN, Tim ; FX ; KAMINSKY, Dan ; GRAND, Joe ; PFEIF, Ken ; DUBRAWSKY, Ido ; BURNETT, Mark ; CRAIG, Paul: *Stealing the Network - How to Own the Box*. Syngress, June 2003. – ISBN 1931836876

[Sandhu u. a. 1996] SANDHU, Ravi S. ; COYNE, Edward J. ; FEINSTEIN, Hal L. ; YOUMAN, Charles E.: Role-Based Access Control Models. In: *IEEE Computer* 29 (1996), Nr. 2, S. 38–47. – URL http://citeseer.nj.nec.com/article/sandhu96rolebased.html. – visited 01.03.2004

[Schneier 1999] SCHNEIER, Bruce: Modeling security threats. In: *Dr. Dobb's Journal* (1999), December. – URL http://www.counterpane.com/attacktrees-ddj-ft.html. – visited 01.03.2004

[Schneier 2000] SCHNEIER, Bruce: Semantic Attacks: The Third Wave of Network Attacks. In: *CRYPTO-GRAM Newsletter* (2000), October. – visited 01.03.2004

[Stahlberg 2000] STAHLBERG, Mika: *Radio Jamming Attacks Against Two Popular Mobile Networks*. Helsinki University of Technology. 2000. – URL http://citeseer.nj.nec.com/401077.html. – visited 01.03.2004

[Tanenbaum 2001] TANENBAUM: *Modern Operating Systems (2nd Edition)*. Prentice Hall, 2001. – 592–600 S. – ISBN 0130313580

[Thaller 1993] THALLER: *Computersicherheit*. Vieweg Verlag, 1993. – 76–104 S. – ISBN 3-528-05372-0

[Thomas und Hunt 1999] THOMAS, Dave ; HUNT, Andy: *The Pragmatic Programmer: From Journeyman to Master*. Addison Wesley, 1999. – ISBN 020161622X

[Wobst 1997] WOBST, Reinhard: *Abenteuer Kryptologie - Methoden, Risiken und Nutzen der Datenverschlüsselung*. Addison Wesley, 1997. – ISBN 3-8273-1193-4

Hiermit versichere ich, dass ich die vorliegende
Arbeit im Sinne der Prüfungsordnung nach §24(4) bzw.
§25(4) ohne fremde Hilfe selbstständig verfasst und nur die
angegebenen Hilfsmittel benutzt habe.

Hamburg 20.04.2004