Mario Pac

Eine Sicherheitsarchitektur für den Einsatz von Chipkarten in Wireless-LAN-Umgebungen

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. rer. nat. Christoph Klauck

Abgegeben am 30. Juni 2004

Mario Pac

Thema der Diplomarbeit

Eine Sicherheitsarchitektur für den Einsatz von Chipkarten in Wireless-LAN-Umgebungen

Stichworte

Wireless-LAN, Chipkarte, Risikoanalyse, Sicherheitsarchitektur, EAP-Verfahren

Kurzzusammenfassung

Das Wireless-LAN ist das meist verbreitete kabellose Netzwerksystem, wobei das bisherige Standard-Protokoll zum Schutz der Wireless-LAN-Kommunikation (WEP) mittlerweile vollständig kompromittiert ist. Eine anerkannte Technologie zur Realisierung von Verschlüsselungs-, Authentifizierungs- und Signierungsanforderungen in sicherheitsrelevanten Bereichen sind Chipkarten. In dieser Arbeit wird daher eine Sicherheitsarchitektur entworfen, die den Einsatz von Chipkarten in Wireless-LAN-Umgebungen ermöglicht. Mittels einer Risikoanalyse werden zunächst konkrete Sicherheitsanforderungen ermittelt. Zur Erfüllung der Anforderungen wird als Basis die RSN-Sicherheitsarchitektur verwendet und darauf aufbauend ein neues EAP-Authentifizierungsverfahren EAP-CHIP entworfen, um die Chipkarten-Technologie in die Sicherheitsarchitektur zu integrieren. Mittels der BAN-Logik wird anschließend die sicherheitstechnische Qualität des neuen Protokolls EAP-CHIP überprüft.

Mario Pac

Title of the paper

A security architecture for the use of smart cards in Wireless LAN environments

Keywords

Wireless LAN, smart card, risk analysis, security architecture, EAP method

Abstract

Wireless LAN is the most usually cable-free network system, whereby the previous standard protocol for the protection of Wireless LAN communication (WEP) are meanwhile completely compromised. An accepted technology for the realization of coding, authentication and signification requirements within security-relevant domains, are smart cards. In this work a security architecture is therefore drafted which makes the use of smart cards in wireless LAN environments possible. With the help of a risk analysis first concrete security requirements are determined. For the completion of the requirements the RSN security architecture is used as basis whereupon constructing the new EAP authentication method EAP-CHIP, in order to integrate the smart card technology into the security architecture. After that the security-relevant quality of new protocol EAP-CHIP is verified with the help of the BAN logic.

Inhaltsverzeichnis

Tabellenverzeichnis **5**

Abbildungsverzeichnis **6**

1 Einleitung **7**
1.1 Motivation . . . 7
1.2 Zielsetzung . . . 7
1.3 Gliederung der Arbeit . . . 8

2 Grundlagen **9**
2.1 Schutzziele . . . 9
2.2 Wireless-LAN . . . 10
2.3 WEP-Sicherheitsarchitektur . . . 12
2.3.1 Authentifizierung . . . 12
2.3.2 Verschlüsselung . . . 13
2.3.3 Entschlüsselung . . . 14
2.3.4 Sicherheitsprobleme . . . 14
2.4 RSN-Sicherheitsarchitektur . . . 16
2.4.1 Authentifizierung . . . 16
2.4.2 Schlüsselmanagement . . . 20
2.4.3 Verschlüsselungsprotokolle . . . 21
2.5 Chipkarten . . . 27
2.5.1 Chipkartenarten . . . 28
2.5.2 PIN-Verfahren . . . 30
2.5.3 Authentifizierung . . . 31
2.5.4 Secure Messaging . . . 33
2.6 EAP-SIM . . . 34
2.6.1 Authentifizierung . . . 35
2.6.2 Sicherheitsmerkmale . . . 36

3 Analyse eines Beispielszenarios 38
3.1 Szenario . . . 38
3.2 Generelle Anforderungen . . . 39
3.3 Risikoanalyse . . . 39
3.3.1 Unberechtigter Netzwerkzugang . . . 40
3.3.2 Abhören der Kommunikationsdaten . . . 42
3.3.3 Verändern der Kommunikationsdaten . . . 42
3.3.4 Verursachen einer Überlastung der Kommunikation . . . 43
3.4 Sicherheitsstrategie . . . 43

4 Entwurf der Sicherheitsarchitektur 45
4.1 Sicherheitsarchitektur . . . 46
4.2 EAP-CHIP-Verfahren . . . 48
4.2.1 Authentifizierungsablauf . . . 48
4.2.2 Optimierung des Nachrichtenablaufs . . . 52

5 Verifizierung des Entwurfs 56
5.1 BAN-Logik . . . 56
5.1.1 Notation . . . 56
5.1.2 Ableitungsregeln . . . 57
5.1.3 Idealisiertes Protokoll . . . 59
5.1.4 Ablauf der Analyse . . . 59
5.2 Idealisierung des Protokolls . . . 60
5.3 Protokollannahmen . . . 61
5.4 Protokollziele . . . 62
5.5 Protokollanalyse . . . 63
5.6 Ergebnis der Analyse . . . 69
5.7 Erfahrungsbericht . . . 69

6 Fazit und Ausblick 71

Literaturverzeichnis 72

Abkürzungsverzeichnis 76

Tabellenverzeichnis

4.1 Abhängigkeiten der Nachrichten 53
4.2 Optimierte Nachrichtenfolge 54

5.1 BAN-Operatoren 57
5.2 Idealisiertes EAP-CHIP-Protokoll 60

Abbildungsverzeichnis

2.1 Basic Service Set (BSS) . . . 11
2.2 ESS-Netzwerk . . . 11
2.3 WEP-Authentifizierung . . . 13
2.4 Blockdiagramm der WEP-Verschlüsselung . . . 13
2.5 Blockdiagramm der WEP-Entschlüsselung . . . 14
2.6 Zusammenspiel der Authentifizierungsprotokolle . . . 16
2.7 Beispiel für eine erfolgreiche RSN-Authentifizierung . . . 19
2.8 RSN-Schlüsselgenerierung . . . 20
2.9 Ablauf des 4-way-handshake-Verfahrens . . . 21
2.10 Funktion(Pseudocode): Michael . . . 23
2.11 Funktion(Pseudocode): MichaelBlock . . . 23
2.12 Blockdiagramm der TKIP-Verschlüsselung . . . 24
2.13 Blockdiagramm der TKIP-Entschlüsselung . . . 25
2.14 Blockdiagramm der CCMP-Verschlüsselung . . . 26
2.15 Blockdiagramm der CCMP-Entschlüsselung . . . 27
2.16 Beispiel für einen Kommandoablauf mit „*INTERNAL AUTHENTICATE*“ . . . 31
2.17 Beispiel für einen Kommandoablauf mit „*EXTERNAL AUTHENTICATE*“ . . . 32
2.18 Beispiel für einen Kommandoablauf mit „*MUTUAL AUTHENTICATE*“ . . . 33
2.19 Ablauf einer vollständigen EAP-SIM-Authentifizierung . . . 35

3.1 Grobarchitektur WLAN mit Chipkarte . . . 38
3.2 Angriffsziel: Unberechtigter Netzwerkzugang . . . 41
3.3 Angriffsziel: Abhören der Kommunikationsdaten . . . 42
3.4 Angriffsziele: Verändern der Kommunikationsdaten, Verursachen einer Überlastung der Kommunikation . . . 43

4.1 Sicherheitsarchitektur . . . 47
4.2 Ablauf einer erfolgreichen EAP-CHIP-Authentifizierung . . . 55

1 Einleitung

Drahtlose Netzwerke werden sowohl in der Geschäftswelt als auch im privaten Umfeld immer beliebter. Die Vorteile liegen in den niedrigen Kosten der Netzwerk-Infrastruktur und in der kabellosen Freiheit (Kauffels 2002).

Das meist verbreitetste kabellose Netzwerksystem ist das Wireless-LAN. Da im Wireless-LAN die Daten über Funk übertragen werden, können diese von jedermann empfangen und nachgeahmt werden. Somit sind Sicherheitstechniken notwendig, die sowohl die Zugriffrechte regeln als auch die zu übertragenden Daten sichern.

Im Bereich der Sicherheitstechniken etablierten sich in den letzten Jahren die Chipkarten. Chipkarten werden bereits in den Bereichen des Mobilfunks, des Bezahlwesens, der Verschlüsselung, der Authentifizierung und der Signierung eingesetzt.

1.1 Motivation

Wenn man mit Hilfe von Chipkarten Sicherheit in den verschiedenen Bereichen gewährleisten kann, dann könnten die Chipkarten auch helfen die Sicherheit im Wireless-LAN zu erhöhen. Das sollte besonders für ein Unternehmen vom Vorteil sein, das bereits Chipkarten für andere Zwecke einsetzt.

1.2 Zielsetzung

Das Ziel dieser Arbeit ist der Entwurf einer Sicherheitsarchitektur für das Wireless-LAN unter Verwendung der Chipkartentechnologie. Die notwendigen Sicherheitsanforderungen sind aus einem Beispielszenario zu ermitteln. Es ist zusätzlich zu verifizieren, ob der Entwurf die Sicherheit gewährleisten kann.

1.3 Gliederung der Arbeit

Nach der Einführung werden zunächst die notwendigen Grundlagen aufgezeigt. Anschließend wird anhand eines Beispielszenarios eine Risikoanalyse durchgeführt, um die Sicherheitsanforderungen zu ermitteln. Dann wird eine Sicherheitsarchitektur mittels der Sicherheitsanforderungen entworfen, was auch den Entwurf eines Authentifizierungsprotokolls beinhaltet. Anschließend wird das entworfene Protokoll durch die BAN-Logik [1] verifiziert. Zum Schluss werden die Erkenntnisse der Arbeit zusammengefasst und ein Ausblick auf eine mögliche Weiterentwicklung gegeben.

[1] Die Abkürzung BAN setzt sich aus den ersten Buchstaben der Nachnamen der Erfinder Burrows, Abadi und Needham zusammen.

2 Grundlagen

In diesem Kapitel werden die Grundlagen aufgezeigt, die für das Verständnis der Arbeit notwendig sind.

2.1 Schutzziele

Die zu schützenden Güter sind für sichere Systeme nach Eckert (2003) die Informationen und Daten. Es gibt Schutzziele, die erreicht werden müssen, um die Sicherheit dieser Informationen bzw. Daten zu gewährleisten. Es werden hier die Schutzziele vorgestellt, die für diese Arbeit relevant sind (vgl. Eckert 2003, S. 6).

Authentizität „Unter der Authentizität eines Objekts bzw. Subjekts (engl. *authenticity*) verstehen wir die Echtheit und Glaubwürdigkeit des Objekts bzw. Subjekts, die anhand seiner eindeutigen Identität und seiner charakterisierenden Eigenschaften überprüfbar ist“(Eckert 2003, S. 6).

Die Prüfung der Authentizität erfolgt durch eine Authentifikation (engl. authentication). Dafür muss bewiesen werden, dass die behauptete Identität des Objekts oder Subjekts mit dessen charakterisierenden Eigenschaften übereinstimmt.

Datenintegrität „Wir sagen, dass das System die Datenintegrität (engl. *integrity*) gewährleistet, wenn es Subjekten nicht möglich ist, die zu schützenden Daten unautorisiert und unbemerkt zu manipulieren“(Eckert 2003, S. 7).

Die Datenintegrität wird durch die Vergabe von Rechten zur Nutzung der Daten gewährleistet. Dazu gehören z. B. Lese- und Schreibrechte für Dateien. Ein anderer Schutz der Datenintegrität ist die Erkennung von Datenmanipulationen, z. B. bei Datenübertragungen. Dies wird durch Hashfunktionen realisiert, die kryptographisch sicher sind.

Informationsvertraulichkeit „Wir sagen, dass das System die Informationsvertraulichkeit (engl. *confidentiality*) gewährleistet, wenn es keine unautorisierte Informationsgewinnung ermöglicht“(Eckert 2003, S. 8).

Die Informationsvertraulichkeit der Daten wird durch den Einsatz von Berechtigungskontrollen sichergestellt, die verhindern, dass unautorisierte Subjekte an diese Informationen gelangen. Die Informationsvertraulichkeit kann auch durch eine Verschlüsselung der Daten gewährleistet werden, so dass kein Subjekt ohne entsprechenden Schlüssel an die Informationen gelangt.

Verfügbarkeit „Wir sagen, dass das System die Verfügbarkeit (engl. *availability*) gewährleistet, wenn authentifizierte Subjekte in der Wahrnehmung ihrer Berechtigungen nicht unautorisiert beeinträchtigt werden können“(Eckert 2003, S. 10).

Eine Gefährdung der Verfügbarkeit kann z. B. durch ein gezieltes Monopolisieren der CPU erfolgen, was dazu führt, dass keine anderen Prozesse mehr ausgeführt werden können. Eine Maßnahme in diesem Fall wäre die Einführung von Quoten, die Systemressourcen wie die CPU-Zeit regeln (vgl. Eckert 2003, S. 6-12).

2.2 Wireless-LAN

Wireless-LAN ist ein drahtloses lokales Netzwerk auf der Basis von Funkübertragungen. Im Jahr 1997 definierte das *Institute of Electrical and Electronics Engineers* (IEEE) den Standard IEEE 802.11, der Möglichkeiten bietet, mit geringem Aufwand drahtlose lokale Netzwerke aufzubauen.

Mittlerweile werden in Flughäfen, Gaststätten, Bahnhöfen usw. so genannte Hot Spots angeboten, die z. B. mobilen Kunden die Internet-Nutzung ermöglichen (vgl. BSI 2002, S. 4).

Die Architektur für das Wireless-LAN wird in IEEE (1999) beschrieben. Alle im Wireless-LAN befindlichen adressierbaren Geräte sind Stationen (STA). Diese Geräte sind mobil, d.h. sie behalten den Zugriff auf das Netzwerk, während sie ihren Standort ändern.

Ein *Basic Service Set* (BSS) ist eine Zusammenfassung von mobilen Geräten, deren Reichweiten sich jeweils überschneiden. Ein BSS besteht aus mindestens 2 mobilen Geräten. Die Abbildung 2.1 zeigt ein BSS, das aus 3 mobilen Geräten besteht.

Wird ein eigenständiges BSS (engl. *independent*) (IBSS) als Netzwerk betrieben, dann wird dieses Netzwerk Ad-hoc-Netzwerk genannt. Ein mobiles Gerät gehört nur solange zum Netzwerk, wie es sich im Bereich des jeweiligen BSS befindet.

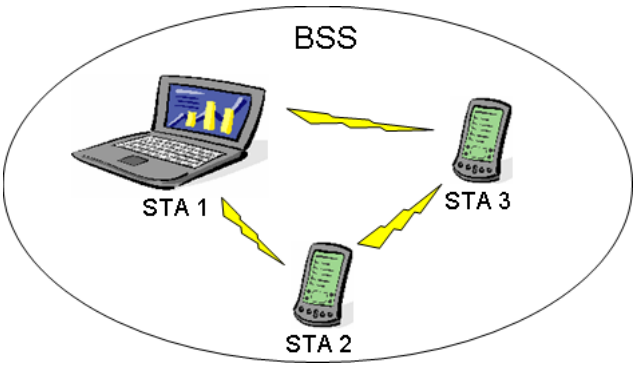

Abbildung 2.1: Basic Service Set (BSS)

Durch den Einsatz eines *Distribution Systems* (DS) ist es möglich, mehrere BSS miteinander zu verbinden. Die Kommunikation, zwischen den mobilen Geräten innerhalb eines BSS und dem DS erfolgt über einen Access-Point (AP). Jedes BSS, das mit dem DS verbunden ist, ist Bestandteil der *Extended Service Set* (ESS). Der ESS bildet zusammen mit dem DS ein ESS-Netzwerk, wie in Abbildung 2.2 dargestellt. Nur mit einem ESS-Netzwerk kann das Wireless-LAN flächendeckend realisiert werden (vgl. IEEE 1999, S. 9-14).

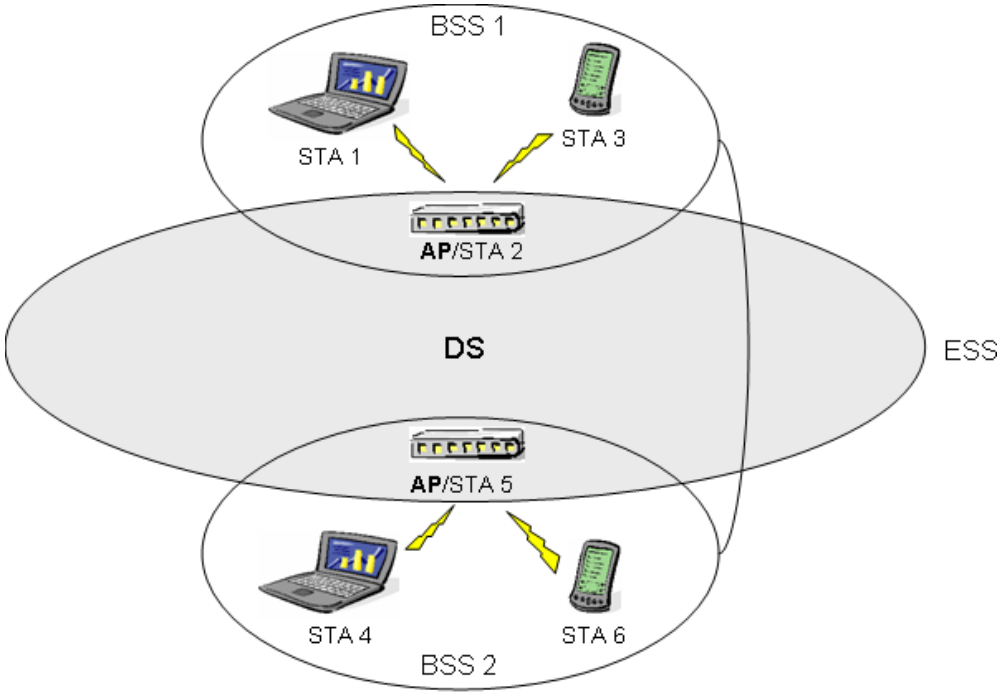

Abbildung 2.2: ESS-Netzwerk

2.3 WEP-Sicherheitsarchitektur

Mit dem Standard IEEE 802.11 (IEEE 1999) werden Sicherheitsmechanismen zur Verschlüsselung, Integritätswahrung und Authentifizierung festgelegt. Diese Sicherheitsmechanismen werden unter den Begriff *Wired Equivalent Privacy* (WEP) zusammengefasst. Ein durch WEP geschütztes Wireless-LAN soll die Schutzziele Authentizität, Datenintegrität und Informationsvertraulichkeit erfüllen.

Die Authentizität des mobilen Geräts wird durch ein Challenge-Response-Verfahren (Beutelspacher u. a. 2004, S. 26-28) geprüft, die Datenintegrität wird durch eine CRC-Prüfsumme (*Cyclic Redundancy Check*) geschützt und die Informationsvertraulichkeit soll durch die Stromchiffre RC4 (Schneider 1996, S. 455-456)[1] gewährleistet werden.

WEP arbeitet mit einem symmetrischen Schlüssel (engl. *shared key*), der sowohl dem mobilen Gerät als auch dem Access-Point bekannt sein muss. Dieser Schlüssel wird sowohl für die Authentifizierung als auch für die Verschlüsselung verwendet.

2.3.1 Authentifizierung

Der Standard unterstützt die Authentifizierungsverfahren *Open System* und *Shared Key*.

Open System Das *Open-System*-Verfahren enthält keine Sicherheitsmechanismen. Es wird nur jeweils eine Nachricht zwischen dem mobilen Gerät und dem Access-Point ausgetauscht. Das mobile Gerät sendet sein Identifikationsmerkmal, und der Access-Point sendet eine Antwort mit dem Ergebnis der Authentifizierung (vgl. IEEE 1999, S. 59).

Shared Key Das *Shared-Key*-Verfahren arbeitet mit einem symmetrischen Schlüssel und wird mittels vier Nachrichten durchgeführt, wie es in Abbildung 2.3 dargestellt wird. Zunächst sendet das mobile Gerät sein Identitätsmerkmal ID_M an den Access-Point. Der Access-Point erstellt einen Zufallswert N_A, der als *Challenge* an das mobile Gerät gesendet wird. Das mobile Gerät verschlüsselt N_A mit dem symmetrischen Schlüssel K und sendet das Ergebnis $\{N_M\}_K$[2] als *Response* zurück an den Access-Point. Der Access-Point entschlüsselt mit dem Schlüssel K und prüft N_A. Abschließend sendet der Access-Point das Ergebnis der Authentifizierung als Antwort A an das mobile Gerät (vgl. IEEE 1999, S. 60-61).

[1] Das Projekt von Grogans u. a. (2000) gibt detaillierte Informationen über RC4.

[2] Eine Verschlüsselung wird durch geschweifte Klammern dargestellt, wobei der Schlüssel tiefer gestellt nachfolgt (z. B. $\{A, B, C\}_K$).

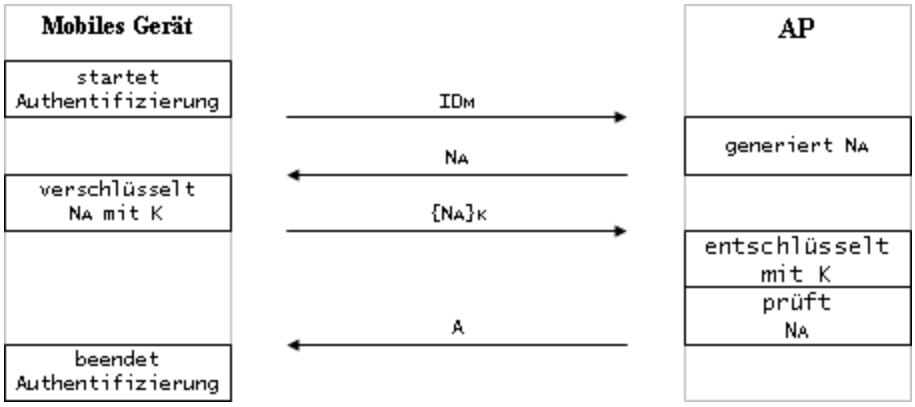

Abbildung 2.3: WEP-Authentifizierung

2.3.2 Verschlüsselung

Zur Verschlüsselung einer Nachricht wird sowohl die Nachricht selbst, der symmetrische Schlüssel als auch ein zufälliger 24-Bit-langer Initialisierungsvektor (IV) benötigt. Der Initialisierungsvektor und der Schlüssel bilden zusammen die Eingabe für den RC4-Algorithmus, der einen Schlüsselstrom liefert. Aus der Klartextnachricht wird eine CRC-Prüfsumme gebildet und als *Integrity Check Value* (ICV) an die Nachricht angefügt. Die anschließende XOR-Verknüpfung mit dem Schlüsselstrom ergibt die verschlüsselte Nachricht. Diese verschlüsselte Nachricht wird zusammen mit dem unverschlüsselten Initialisierungsvektor gesendet. Die Abbildung 2.4 zeigt die Verschlüsselung als Blockdiagramm (vgl. IEEE 1999, S. 62-63).

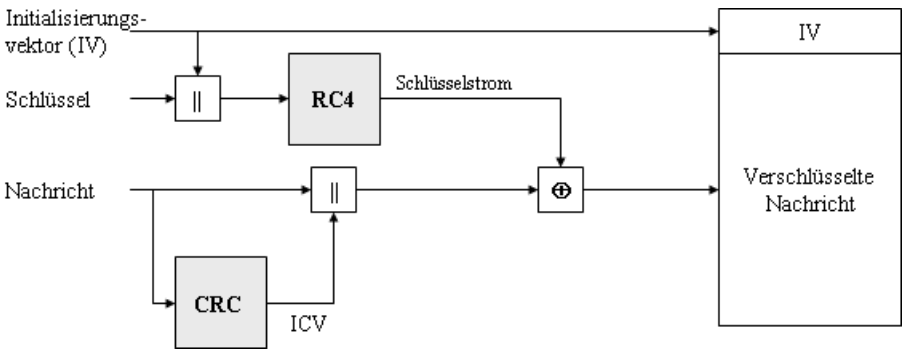

Abbildung 2.4: Blockdiagramm der WEP-Verschlüsselung (vgl. IEEE 1999, S. 63)

2.3.3 Entschlüsselung

Zum Entschlüsseln werden die verschlüsselte Nachricht, der Initialisierungsvektor und der symmetrische Schlüssel benötigt. Der Initialisierungsvektor und der Schlüssel werden zusammengefügt und als Eingabe an den RC4-Algorithmus gegeben, der einen Schlüsselstrom generiert. Dieser wird mit der verschlüsselten Nachricht XOR-verknüpft, wodurch die Nachricht entschlüsselt wird. Nun wird eine neue CRC-Prüfsumme ICV' gebildet. Wenn der empfangene ICV gleich dem ICV' ist, dann war die Entschlüsselung erfolgreich. Die Abbildung 2.5 zeigt das entsprechende Blockdiagramm für die Entschlüsselung (vgl. IEEE 1999, S. 63-64).

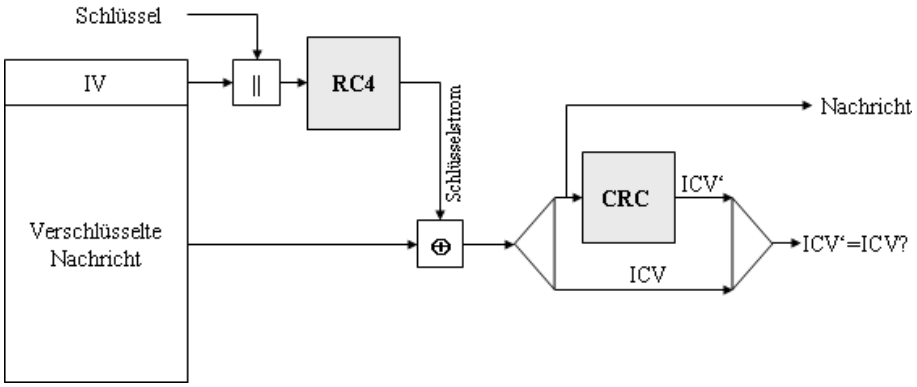

Abbildung 2.5: Blockdiagramm der WEP-Entschlüsselung (vgl. IEEE 1999, S. 64)

2.3.4 Sicherheitsprobleme

Laut BSI (2002) wurde WEP bis heute vollständig kompromittiert. Die Schutzziele Authentizität, Datenintegrität und Informationsvertraulichkeit werden nicht erreicht. Es gibt frei verfügbare Tools für passive Angriffe auf ein WEP-geschütztes Wireless-LAN. Nachfolgend werden einige der von BSI (2002) beschriebenen Sicherheitsprobleme aufgezeigt (vgl. BSI 2002, S. 8-13).

Fehlendes Schlüsselmanagement

In jedem mobilen Gerät und Access-Point muss der symmetrische Schlüssel von Hand eingetragen werden. Da für jedes mobile Gerät derselbe Schlüssel verwendet wird, wird durch das bekannt werden des Schlüssels das gesamte Wireless-LAN kompromittiert (vgl. BSI 2002, S. 9).

Initialisierungsvektor zu kurz

Der Initialisierungsvektor ist dazu da, damit für jede Verschlüsselung ein neuer Schlüsselstrom erzeugt wird. Sobald zwei Datenpakete mit demselben Initialisierungsvektor auftreten, lassen sich daraus meist sowohl die Datenpakete als auch der Schlüsselstrom rekonstruieren. Bei 24 Bit können 16,8 Mio. verschiedene Initialisierungsvektor erzeugt werden. Wenn der Initialisierungsvektor zufällig erzeugt wird, dann tritt nach ca. 4000 Datenpaketen ein Initialisierungsvektor wiederholt auf. Bei einem hohen Datenaufkommen ist wahrscheinlich nach einer Stunde jeder mögliche Initialisierungsvektor einmal aufgetreten und der weitere Datenverkehr kann relativ zuverlässig mitgelesen werden (vgl. BSI 2002, S. 10).

Fälschen von Datenpaketen

Der Schlüsselstrom wird anhand des Initialisierungsvektors und des Schlüssels generiert. Bekommt ein Angreifer einen solchen Schlüsselstrom in den Besitz, dann kann er, solange der Schlüssel gültig ist, beliebige Datenpakete fälschen, also als „korrekte“ Datenpakete einschleusen. Das ist möglich, weil der Initialisierungsvektor vom Sender festgelegt wird und ein Initialisierungsvektors beliebig oft verwendet werden darf (vgl. BSI 2002, S. 10).

Brechen der Authentifizierung

Wenn ein Angreifer eine vollständige Authentifizierung aufzeichnet, dann ist er in der Lage, sich zu jedem Zeitpunkt erfolgreich zu authentifizieren, ohne dass er im Besitz des Schlüssels ist. Dazu muss er die *Response* mit der *Challenge* XOR-verknüpfen. Dadurch erhält er den Schlüsselstrom, der für eine eigene Authentifizierung eingesetzt werden kann. Da sowohl für die Authentifizierung als auch für die Verschlüsselung derselbe Schlüssel verwendet wird, können mit diesem Schlüsselstrom auch beliebige Datenpakete gefälscht werden (vgl. BSI 2002, S. 10).

Integritätsprüfung wirkungslos

Durch die CRC-Prüfsumme sollen mutwillige Veränderungen der Datenpakete erkannt werden. CRC wurde entwickelt, um Daten vor zufälligen Veränderungen zu schützen. Auf Grund der Linearität der CRC-Prüfsumme und der XOR-Struktur des Stromchiffrier-Algorithmus ist es einem Angreifer möglich, die Daten und die CRC-Prüfsumme so zu manipulieren, dass eine Veränderung nicht erkennbar ist (vgl. BSI 2002, S. 10).

2.4 RSN-Sicherheitsarchitektur

Aufgrund der Sicherheitsprobleme des WEP (vgl. Kapitel 2.3.4) entwirft das IEEE zurzeit eine neue Sicherheitsarchitektur, die im Standard IEEE 802.11i (IEEE 2003) festgelegt wird. Der Nachfolger von WEP wird *Robust Security Network* (RSN) genannt. Der Standard befindet sich noch in der Entwicklung. Der Arbeit liegt der *Draft 4.1* zu Grunde.

2.4.1 Authentifizierung

Das mobile Gerät muss sich gegenüber einem Authentication-Server (AS) über das *Extensible Authentication Protocol* (EAP) authentifizieren. Da das mobile Gerät nicht direkt mit dem Authentication-Server kommunizieren kann, muss der Access-Point als Übermittler der EAP-Nachrichten dienen. Zur Übertragung dieser Nachrichten zwischen dem mobilen Gerät und dem Access-Point ist nach IEEE (2003) das Protokoll *EAP over LAN* (EAPOL) zu verwenden. Dieses Protokoll ist im Standard IEEE 802.1X (IEEE 2003) definiert. Zwischen dem Access-Point und dem Authentication-Server müssen die EAP-Nachrichten ebenfalls übertragen werden. Hier wird von der RSN-Sicherheitsarchitektur nicht vorgeschrieben, welches Protokoll zu verwenden ist. Da im Standard IEEE 802.11i der *Remote Authentication Dial In User Service* (RADIUS) erwähnt wird, und dieser nach Aboba u. a. (2003) die Möglichkeit bietet, EAP-Nachrichten zu übertragen, wird auch in dieser Arbeit RADIUS verwendet. Die Abbildung 2.6 zeigt die erwähnten Protokolle im Zusammenhang mit den beteiligten Komponenten.

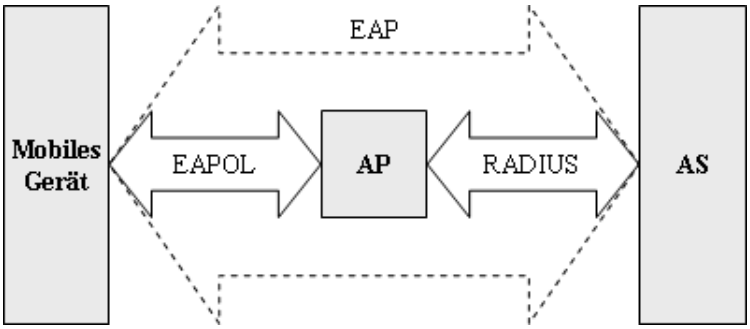

Abbildung 2.6: Zusammenspiel der Authentifizierungsprotokolle

EAP

Ursprünglich wurde EAP zur Ergänzung des *Point-to-Point Protocols* (PPP) entwickelt. Das PPP wurde für Einwahl-Netzwerke entwickelt, und kannte bis dahin nur die Authentifizie-

rungsverfahren PAP (*Password Authentication Protocol*) und CHAP (*Challenge Handshake Authentication Protocol*). Durch das EAP wurde das PPP erweiterbar (engl. *extensible*) gemacht, was die Verwendung von beliebigen Authentifizierungsverfahren für das PPP ermöglicht. EAP wurde in Blunk u. a. (1998) spezifiziert.

Eine EAP-Authentifizierung erfolgt zwischen einem Client und einem Authentifikator. Dabei schickt der Authentifikator *EAP-Request*-Nachrichten an den Client, der diese mittels *EAP-Response*-Nachrichten beantworten muss. Diese Nachrichten werden durch Typnummern spezifiziert. Sendet der Authentifikator z. B. eine *Request*-Nachricht mit der Typnummer 1, dann verlangt dieser die Bezeichnung der Identität (engl. *Identity*), und der Client muss eine *EAP-Response*-Nachricht mit der Typnummer 1 und seiner Identitätsbezeichnung zurücksenden. Die Typnummern 1 bis 6 sind reserviert und in Blunk u. a. (1998) beschrieben. Alle weiteren Typnummern können von EAP-Verfahren belegt werden. Die *Internet Assigned Numbers Authority* (IANA) verwaltet die Zuordnung der Typnummern zu den bekannten EAP-Verfahren; vgl. IANA (2004). Zum Abschluss einer erfolgreichen Authentifizierung sendet der Authentifikator eine *EAP-Success*-Nachricht an den Client. Sollte dagegen die Authentifizierung fehlgeschlagen sein, dann wird diese mit einer *EAP-Failure*-Nachricht vom Authentifikator abgeschlossen.

EAPOL

Der Standard IEEE 802.1X ermöglicht das Austauschen von EAP-Nachrichten zwischen einem Gerät und einem Authentifikator über ein beliebiges LAN. In IEEE (2001) wird beschrieben, wie diese Nachrichten in EAPOL-Nachrichten gekapselt werden (vgl. IEEE 2001, S. 13 ff.). Das EAPOL kennt die folgenden 4 Nachrichtenarten:

EAPOL-Start Mit dem Senden einer EAPOL-Start-Nachricht an einen Authentifikator wird von einem Gerät eine Authentifizierung initiiert.

EAP-Packet In *EAP-Packet*-Nachrichten werden die eigentlichen EAP-Nachrichten so gekapselt, dass sie über ein LAN übertragen werden können.

EAPOL-Logoff Mit der *EAPOL-Logoff*-Nachricht teilt das Gerät dem Authentifikator mit, dass es die Authentifizierung beenden möchte.

EAPOL-Key Der Nachrichtentyp *EAPOL-Key* dient zum Austausch von Schlüsselinformationen zwischen dem Gerät und dem Authentifikator. Dieser Nachrichtentyp steht nicht im direkten Zusammenhang mit den EAP-Nachrichten.

EAPOL-Encapsulated-ASF-Alert Mit einer solchen Nachricht werden Fehler gemeldet.

RADIUS

Mit RADIUS wird in Rigney u. a. (1997) ein Protokoll spezifiziert, das es ermöglicht, Authentifizierungen, Autorisierungen und Austausch von Konfigurationsinformationen zwischen Netzzugangsservern und einen zentralisierten Authentication-Server durchzuführen. Das RADIUS-Protokoll kennt vier verschiedene Nachrichtenarten. Zusätzlich wird in Aboba u. a. (2003) beschrieben, wie diese Nachrichtenarten für den Transport von EAP-Nachrichten verwendet werden können.

Access-Request Eine solche Nachricht wird an einen Authentication-Server gesendet und enthält Informationen über den Benutzer, der Zugriff auf einen Netzzugangsserver haben möchte. In Verbindung mit EAP ist diese Nachrichtenart dafür zuständig, *EAP-Response*-Nachricht an den Authentication-Server zu senden.

Access-Challenge Der Authentication-Server sendet eine solche Nachricht, wenn er vom Benutzer eine Antwort auf ein *Challenge* benötigt. Diese Nachrichtenart kann *EAP-Request*-Nachrichten transportieren.

Access-Accept Der Authentication-Server teilt mit einer solchen Nachricht mit, dass der angefragte Zugriff gestattet ist. Mit solchen Nachrichten werden *EAP-Success*-Nachrichten übermittelt. Zusätzlich kann mit dieser Nachricht ein Sitzungsschlüssel mittels des *MS-MPPE-RECV-Key*-Attributs übermittelt werden. Das *MS-MPPE-RECV-Key*-Attribut ist ein RADIUS-Attribut, das in Zorn und Corporation (1999) beschrieben ist. Die MS-MPPE-Attribute ermöglichen die Unterstützung des *Microsoft Point-to-Point Encryption Protocol* (MPPE) unter RADIUS.

Access-Reject Mit einer Access-Reject-Nachricht lehnt der Authentication-Server die Zugriffsanfrage ab. Diese Nachrichtenart kann die *EAP-Failure*-Nachricht transportieren.

Ablauf der Authentifizierung

Anhand der zuvor beschriebenen Protokolle und ihrer Nachrichten wird nun ein Beispiel eines möglichen Nachrichtenaustauschs veranschaulicht. Der Austausch findet während einer erfolgreichen Authentifizierung innerhalb eines durch das RSN geschützten Wireless-LAN statt.

Es wird davon ausgegangen, dass ein RADIUS-Server als Authentication-Server eingesetzt wird. Weiterhin wird kein spezielles EAP-Verfahren ausgewählt. Es wird nur angenommen, dass das EAP-Verfahren einen Sitzungsschlüssel zwischen dem mobilen Gerät und dem Authentication-Server (AS) vereinbart. Dieser Sitzungsschüssel muss zum Ende der erfolgreichen Authentifizierung vom Authentication-Server an den Access-Point übertragen werden.

Die Authentifizierung beginnt mit einer *EAPOL-Start*-Nachricht, diese wird vom mobilen Gerät zum Access-Point gesendet. Zur Ermittlung der Identität des mobilen Geräts sendet der Access-Point eine *EAP-Request*-Nachricht des Typs 1 (*Identity*), verpackt in einer *EAP-Packet*-Nachricht, an das mobile Gerät. Das mobile Gerät beantwortet diese Anfrage mit der entsprechenden *EAP-Response*-Nachricht, die ebenfalls in einer *EAP-Packet*-Nachricht verpackt wird. Der Access-Point ermittelt aus der Antwort die *EAP-Response*-Nachricht und sendet diese, eingebettet in einer *Access-Request*-Nachricht, an den Authentication-Server. Nun kennt der Authentication-Server das mobile Gerät und er kann eine EAP-Authentifizierung durchführen. Dafür sendet er *EAP-Request*-Nachrichten, verpackt in *Access-Challenge*-Nachrichten an den Access-Point. Der Access-Point leitet die *EAP-Request*-Nachricht, in einer *EAP-Packet*-Nachricht verpackt, weiter an das mobile Gerät. Das mobile Gerät sendet die entsprechende *EAP-Response*-Nachricht, eingebettet in einer *EAP-Packet*-Nachricht, an den Access-Point. Der Access-Point sendet diese *EAP-Response*-Nachricht, in einer *Access-Request*-Nachricht, weiter an den Authentication-Server. Nach jeder erhaltenen *EAP-Response*-Nachricht kann der Authentication-Server eine weitere *EAP-Request*-Nachricht senden. Sobald das EAP-Verfahren erfolgreich war, wird eine *EAP-Success*-Nachricht in einer *Access-Accept*-Nachricht verpackt. Der Authentication-Server sendet diese Nachricht zusammen mit dem Sitzungsschlüssel als *MS-MPPE-RECV-Key*-Attribut, an den Access-Point. Der Access-Point ermittelt den Sitzungsschlüssel aus dem *MS-MPPE-RECV-Key*-Attribut und leitet die *EAP-Success*-Nachricht weiter an das mobile Gerät, verpackt in einer *EAP-Packet*-Nachricht. Die Abbildung 2.7 zeigt diesen beschriebenen Nachrichtenablauf.

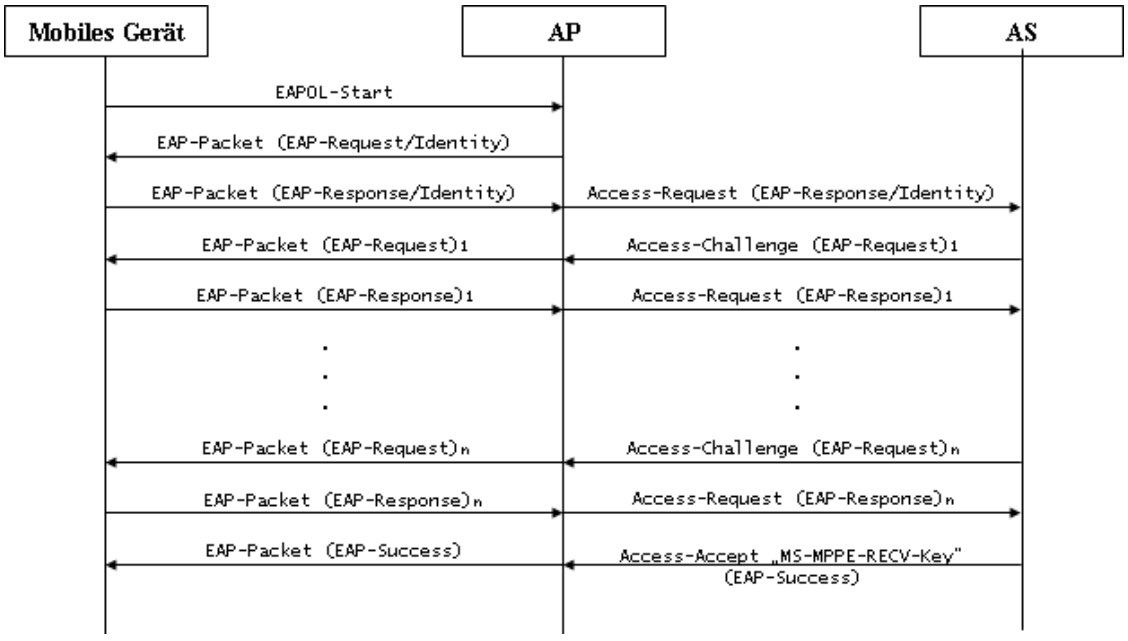

Abbildung 2.7: Beispiel für eine erfolgreiche RSN-Authentifizierung

2.4.2 Schlüsselmanagement

Nach einer erfolgreichen Authentifizierung erhalten sowohl das mobile Gerät, als auch der Access-Point einen zufälligen symmetrischen Sitzungsschlüssel, der vom gewählten EAP-Verfahren festgelegt wurde. Dieser Schlüssel wird in der RSN-Sicherheitsarchitektur *Pairwise Master Key* (PMK) genannt. In IEEE (2003) wird festgelegt, dass mit einem *4-way-handshake*-Verfahren nun zwischen dem mobilen Gerät und dem Access-Point ein *Pairwise Transient Key* (PTK) vereinbart wird. Der PTK wird mittels einer festgelegten *Pseudo-Random Function* (PRF) und zwei Zufallswerten (N_M und N_A) bestimmt. Der PTK ist je nach zu verwendenden Verschlüsselungsverfahren (TKIP [*Temporal Key Integrity Protocol*] oder CCMP [*Counter-Mode/CBC-MAC Protocol*]) unterschiedlich lang zu erzeugen. Die ersten 128 Bit werden für die Erzeugung eines *Message Integrity Code* (MIC)[3] einer EAPOL-Nachricht verwendet. Die folgenden 128 Bit sind für die Verschlüsselung der EAPOL-Nachrichten. Bei TKIP folgen weitere 256 Bits für zwei TKIP-Schlüssel. Dagegen folgt bei CCMP nur ein weiterer Schlüssel, der eine Länge von 128 Bit hat. Die Abbildung 2.8 veranschaulicht den Zusammenhang der Schlüssel.

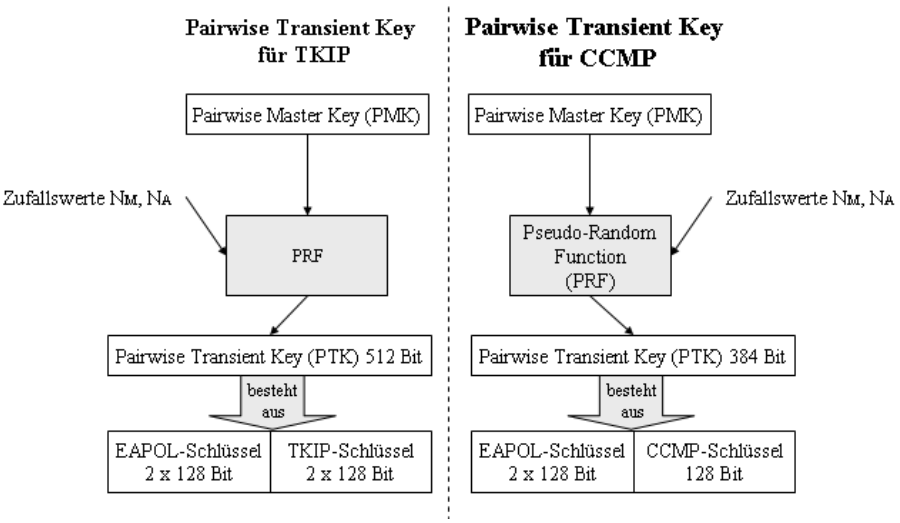

Abbildung 2.8: RSN-Schlüsselgenerierung

Die Abbildung 2.9 zeigt den Ablauf des *4-way-handshake*-Verfahrens. Es kommen nur *EAPOL-Key*-Nachrichten zum Einsatz. Zunächst generiert das mobile Gerät einen Zufallswert N_M und der Access-Point den Zufallswert N_A. Der Access-Point sendet N_A an das mobile Gerät. Das mobile Gerät bestimmt den PTK durch die PRF, mit den Zufallswerten N_M und N_A als Eingabe. Anschließend sendet das mobile Gerät seinen Zufallswert N_M, mit dem MIC der Nachricht, an den Access-Point. Der Access-Point erzeugt ebenfalls den PTK und prüft

[3] Auch als *Message Authentication Code* (MAC) bekannt

den MIC. Dann sendet der Access-Point seinen Wert N_A, mit dem MIC dieser Nachricht, an das mobile Gerät, das den MIC prüft. Als letztes sendet das mobile Gerät nur eine Nachricht mit einem MIC an den Access-Point. Danach legen das mobile Gerät und der Access-Point die Schlüssel für das jeweilige Verschlüsselungsverfahren anhand des PTK fest (vgl. IEEE 2003, S. 73-90).

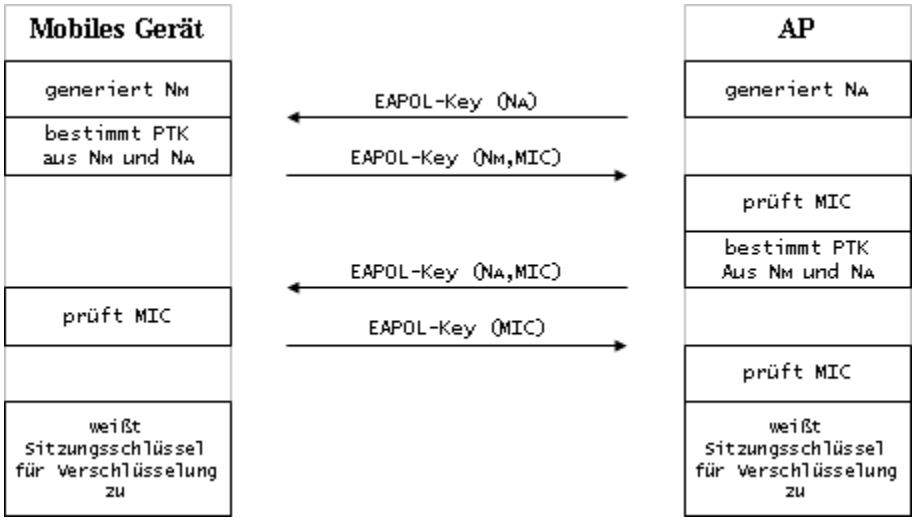

Abbildung 2.9: Ablauf des 4-way-handshake-Verfahrens

2.4.3 Verschlüsselungsprotokolle

Der Standard IEEE 802.11i (IEEE 2003) definiert zwei Datenverschlüsselungsprotokolle: das TKIP und das CCMP. Das TKIP wurde nur dazu entworfen, damit auch Geräte, die WEP-Algorithmen verwenden, unterstützt werden können. Das CCMP soll in Zukunft hauptsächlich in Geräten implementiert sein.

TKIP

Das *Temporal Key Integrity Protocol* (TKIP) ist eine Erweiterung der WEP-Verschlüsselung. Es modifiziert die WEP-Verschlüsselung wie folgt:

1. TKIP verschlüsselt MSDU-Datenpakete (*MAC [Medium Access Control] service data unit*), die in MPDU-Datenpakete (*MAC protocol data unit*) aufgeteilt werden. Diese von IEEE (1999) spezifizierten Formate sind speziell für die Übertragung über das Wireless-LAN konzipiert worden.

2. TKIP generiert einen MIC über das MSDU-Datenpaket. Der MIC wird zusammen mit den Daten verschlüsselt und vom Empfänger nach der Entschlüsselung geprüft.

3. Da ein Angreifer einen MIC mit wenigen Nachrichten kompromittieren kann, arbeitet TKIP mit *countermeasures*. Die *countermeasures* verhindern die Wahrscheinlichkeit einer erfolgreichen Fälschung einer Nachricht. Und sie verringern die Anzahl der Informationen, die der Angreifer für die Ermittlung des Schlüssels benötigt. Die Logik der *countermeasures* ist in IEEE (2003) auf Seite 43 bis 51 beschrieben.

4. TKIP benutzt einen *TKIP sequence counter* (TSC), der *Replay*-Angriffe[4] verhindern soll. Der TSC wird in WEP-Initialisierungsvektor (WEP IV) mit eingebunden, und so zum Empfänger übertragen.

5. TKIP bezieht zur Sicherung der Daten Informationen aus dem MAC-*Header* (IEEE 1999, S. 29-58) mit ein, der Bestandteil von MSDU- bzw. MPDU-Datenpaketen ist. Dabei handelt es sich um die Informationen zur *transmitter address* (TA), *source address* (SA), *destination address* (DA) und *priority*.

6. TKIP verwendet eine Mix-Funktion, um den symmetrischen Schlüssel, die *transmitter address* und den TSC miteinander zu kombinieren. Das Ergebnis wird teilweise als Initialisierungsvektor und teilweise als Eingabe für den RC4-Algorithmus des WEP verwendet. Der Empfänger ermittelt den TSC und kann ebenfalls den Initialisierungsvektor und die Eingabe für den RC4-Algorithmus ermitteln und so die Daten entschlüsseln.

MIC Zum Schutz der Integrität bestimmt TKIP einen MIC, der *Michael* genannt wird. Nach IEEE (2003) bietet *Michael* nur einen schwachen Schutz vor Nachrichtenfälschungen, aber es ist das Beste, was mit der Mehrheit der gegenwärtigen Hardware möglich ist. Nun folgt die Beschreibung des Algorithmus, wie in IEEE (2003) beschrieben.

Durch den *Michael*-Algorithmus wird ein 64-Bit-MIC generiert. Der 64 Bit lange *Michael*-Schlüssel wird in acht 8-Bit-Werten ($k_0, \ldots, k_7$) eingeteilt. Diese werden dann zu zwei 32-Bit-Werten (K_0, K_1) konvertiert. Die Daten, über denen der MIC bestimmt werden soll, werden in die 8-Bit-Sequenz ($m_0, \ldots, m_{n-1}$) eingeteilt, wobei n die Anzahl der 8-Bit-Werte ist und durch 4 teilbar sein muss. Dafür werden den Daten zuvor so viele 8-Bit-Werte mit dem Wert 0 hinzugefügt, bis die Gesamtanzahl durch 4 teilbar ist. Die Daten entsprechen der Kombination von *source address*, *destination address*, *priority* und MSDU-Datenpaket. Dem Ergebnis wird der 8-Bit-Wert $0x5a$ angefügt. Die 8-Bit-Sequenz ($m_0, \ldots, m_{n-1}$) wird in die 32-Bit-Sequenz ($M_0, \ldots, M_{N-1}$) konvertiert, wobei $N = (n + 5/4)$ ist, und N aufgerundet wird.

[4] Bei einem Replay-Angriff wird eine bereits gesendete Nachricht wiederholt eingespielt (vgl. Beutelspacher u. a. 2004, S. 99).

Input: Key (K_0, K_1), Data $(M_0, ..., M_{N-1})$
Output: MIC (V_0, V_1)
  $(l, r) \leftarrow (K_0, K_1)$
  for $i = 0$ to $N - 1$ **do**
    $l \leftarrow l \oplus M_i$
    $(l, r) \leftarrow MichaelBlock(l, r)$
  end for
  return (l, r)

Abbildung 2.10: Funktion(Pseudocode): Michael (vgl. IEEE 2003, S. 42)

Der MIC wird iterativ bestimmt. Gestartet wird mit dem Schlüssel-Wert (K_0, K_1). Für jede 32-Bit-Sequenz wird dann die Blockfunktion $MichaelBlock$ aufgerufen, wie in der Abbildung 2.10 abgebildet. Die Blockfunktion wird N-mal aufgerufen, am Ende enthält (l, r) zwei 32-Bit-Werte, die als Ergebnis (V_0, V_1) von der Funktion als MIC zurückgegeben werden.

Input: (l, r)
Output: (l, r)
  $r \leftarrow r \oplus (l <<< 17)$
  $l \leftarrow (l + r) mod 2^{32}$
  $r \leftarrow r \oplus XSWAP(l)$
  $l \leftarrow (l + r) mod 2^{32}$
  $r \leftarrow r \oplus (l <<< 3)$
  $l \leftarrow (l + r) mod 2^{32}$
  $r \leftarrow r \oplus (l >>> 2)$
  $l \leftarrow (l + r) mod 2^{32}$
  return (l, r)

Abbildung 2.11: Funktion(Pseudocode): MichaelBlock (vgl. IEEE 2003, S. 43)

Die Abbildung 2.11 zeigt die *Michael*-Blockfunktion. Dies ist ein Feistel-type-Konstrukt (Feistel 1973) mit abwechselnden Additions- und XOR-Operationen. Die Operatoren $<<<$ und $>>>$ entsprechen einer bitweisen Rotation nach links bzw. rechts. Die Funktion $XSWAP$ tauscht die Positionen der beiden höchstwertigen 8-Bit-Werte und der beiden niedrigstwertigen 8-Bit-Werte (vgl. IEEE 2003, S. 40-43).

Verschlüsselung Für die TKIP-Verschlüsselung werden das zu verschlüsselnde MSDU-Datenpaket, die *transmitter address*, die *source address*, die *destination address* und die *priority* benötigt. Die notwendigen Schlüssel, TKIP-Schlüssel und MIC-Schlüssel, sind Bestandteile des *Pairwise Transient Key* (PTK).

Mittels des MIC-Schlüssels wird der MIC, über die Kombination aus der *source address*, der *destination address*, der *priority* und aus dem MSDU-Datenpaket, bestimmt. Der MIC wird an das MSDU-Datenpaket angehängt, und das Ergebnis wird in sendefähige MPDU-Datenpakete aufgesplittert (engl. *fragment*). Jedem MPDU-Datenpaket wird ein TSC angefügt, der für jedes MPDU-Datenpaket um eins erhöht wird. Zusätzlich werden, durch einen zweiphasigen Mix, der Initialisierungsvektor und die Eingabe für den RC4-Algorithmus bestimmt. In der ersten Phase werden der TKIP-Schlüssel, die *transmitter address* und der TSC miteinander kombiniert, was den *TKIP mixed Transmit Address and Key* (TTAK) ergibt. In der zweiten Phase wird der TTAK mit dem TSC vermischt, und anschließend in Initialisierungsvektor und Eingabe für RC4-Algorithmus aufgeteilt. Die Mix-Funktionen werden nicht in dieser Arbeit beschrieben, da dies den Umfang der Arbeit sprengen würde. Wie die Mix-Funktionen zu implementieren sind, kann in IEEE (2003) auf Seite 51 bis 55 entnommen werden. Die Abbildung 2.12 zeigt die Verschlüsselung als Blockdiagramm (vgl. IEEE 2003, S. 36-37).

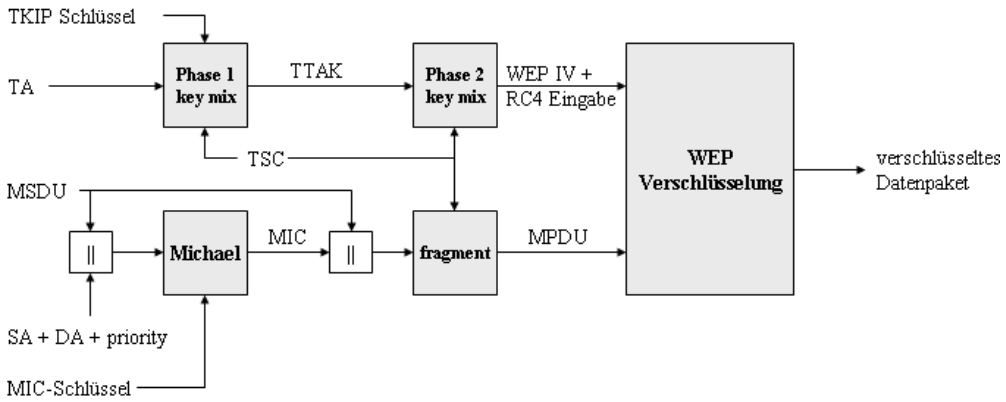

Abbildung 2.12: Blockdiagramm der TKIP-Verschlüsselung (vgl. IEEE 2003, S. 37)

Entschlüsselung Die TKIP-Entschlüsselung beginnt mit der Extrahierung des TSC aus dem Initialisierungsvektor. Anschließend wird geprüft, ob der TSC des Datenpakets entsprechend der Reihenfolge korrekt ist. Dann werden der TKIP-Schlüssel, die *transmitter address* und der TSC miteinander vermischt, was den TTAK ergibt. Nun wird der TTAK und der TSC miteinander vermischt, das ergibt die Eingabe für den RC4-Algorithmus des WEP. Danach kann durch die WEP-Entschlüsselung das entsprechende MPDU-Datenpaket entschlüsselt werden. Die MPDU-Datenpakete werden zusammengesetzt und ergeben so das MSDU-Datenpaket mit dem jeweiligen MIC. Mittels des MIC-Schlüssels wird der MIC des MSDU-Datenpakets neu bestimmt und mit dem übertragenen MIC verglichen. Wenn beide gleich sind, dann war die Entschlüsselung erfolgreich und das MSDU-Datenpaket enthält die entschlüsselte Nachricht (vgl. IEEE 2003, S. 37-38).

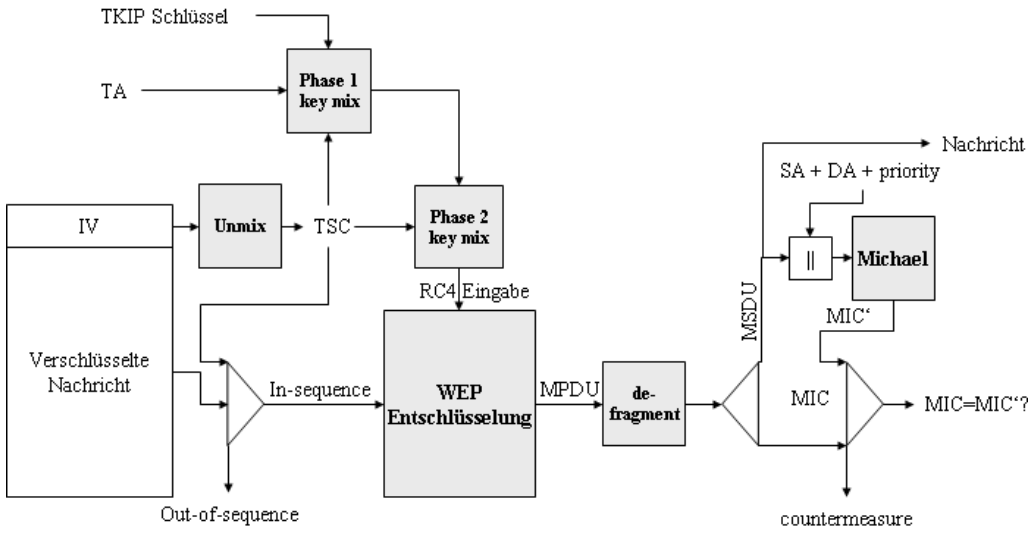

Abbildung 2.13: Blockdiagramm der TKIP-Entschlüsselung (vgl. IEEE 2003, S. 38)

CCMP

Nach IEEE (2003) gewährleistet das *Counter-Mode/CBC-MAC protocol* (CCMP) den Schutz der Authentizität, der Vertraulichkeit, der Integrität und vor *Replay*-Angriffen. CCMP ist vorgeschrieben für die Einhaltung der RSN-Sicherheitsarchitektur.

Das CCMP verwendet den *Advanced Encryption Standard* (AES) (FIPS PUB 197 2001) im CCM-Modus (*Counter with CBC-MAC*). Der CCM-Modus, definiert in Whiting u. a. (2003), verwendet den *Counter Mode* (CTR) zum Schutz der Vertraulichkeit, und den Cipher Block Chaining Message Authentication Code (CBC-MAC) zum Schutz von Authentizität und Integrität. Die AES-Modi sind in Dworkin (2001) dokumentiert. AES ist eine Blockchiffre, die in 128-Bit-Blöcken verschlüsselt. CCMP verwendet für die AES-Verschlüsselung einen 128-Bit-Schlüssel, der als Teil des PTK an das CCMP geliefert wird. Zusätzlich verwendet das CCMP eine Paketnummer (PN), zur Kennzeichnung des MPDU-Datenpakets. Diese Paketnummer ist ein 48-Bit-Wert. Jeder mögliche Wert der Paketnummer darf nur einmal innerhalb einer Verbindung verwendet werden. Zusätzlich bezieht CCMP den ursprünglichen *MAC-Header* in die Verschlüsselung mit ein.

Verschlüsselung Zur Verschlüsselung eines MPDU-Datenpakets werden der CCMP-Schlüssel, die Paketnummer und der im MPDU-Datenpaket enthaltene MAC-*Header* benötigt. Aus dem MAC-*Header* werden zusätzlich die *address 2* (A2) und die *priority* verwendet.

Die Abbildung 2.14 zeigt die CCMP-Verschlüsselung als Blockdiagramm. Als erstes wird die Paketnummer inkrementiert. Mit der Paketnummer wird dann der CCMP-*Header* erstellt.

Anschließend wird ein *Nonce*-Wert abhängig von der Paketnummer, address 2 und *priority* gebildet. Der MAC-*Header* wird 1 zu 1 übernommen und aus den enthaltenen Informationen wird die *Additional Authentication Data* (AAD) erstellt. Die MPDU-Daten, die AAD, der *Nonce*-Wert und der CCMP-Schlüssel dienen nun als Eingabe für die CCM-Verschlüsselung. CCM liefert die verschlüsselten Daten und den MIC. Die MAC-*Header*, die verschlüsselten Daten, der MIC und der CCMP-*Header* bilden zusammen das neue verschlüsselte MPDU-Datenpaket (vgl. IEEE 2003, S. 57-60).

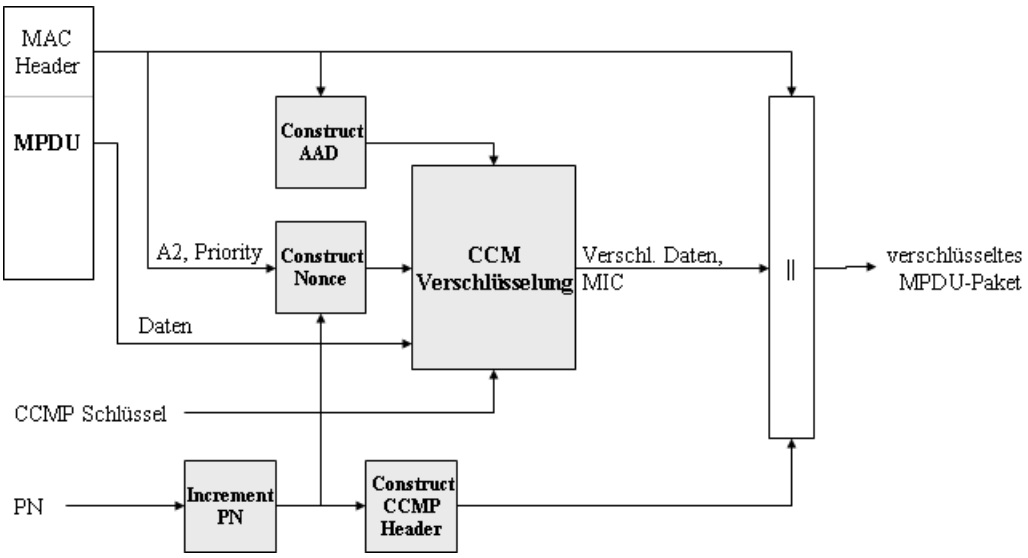

Abbildung 2.14: Blockdiagramm der CCMP-Verschlüsselung (vgl. IEEE 2003, S. 57)

Entschlüsselung Zur Entschlüsselung wird ein mit CCMP verschlüsseltes MPDU-Datenpaket, der CCMP-Schlüssel, eine Paketnummer und der im MPDU-Datenpaket enthaltene MAC-*Header* benötigt. Zusätzlich liefert der MAC-*Header* die address 2 und die *priority*. Die im MPDU-Datenpaket enthaltenen Daten bestehen aus den verschlüsselten Daten und dem MIC. Das MPDU-Datenpaket liefert im CCMP-*Header* eine eigene *Header*-Paketnummer.

Die CCMP-Entschlüsselung wird in der Abbildung 2.15 dargestellt. Anhand der Informationen aus dem MAC-*Header* wird die AAD generiert. Der *Nonce*-Wert wird aus der address 2, der *priority* und der CCMP-*Header*-Paketnummer erstellt. Die MPDU-Daten, der MIC, die AAD, der *Nonce*-Wert und der CCMP-Schlüssel dienen nun als Eingabe für die CCM-Entschlüsselung. Diese liefert die entschlüsselten Daten, die zusammen mit dem MAC-*Header* das unverschlüsselte MPDU-Datenpaket ergeben. Als letztes wird die CCMP-*Header*-Paketnummer mittels der nebenher bestimmt Paketnummer geprüft. Wenn die CCMP-*Header*-Paketnummer korrekt ist, dann ist das MPDU-Datenpaket erfolgreich entschlüsselt worden (vgl. IEEE 2003, S. 57-61).

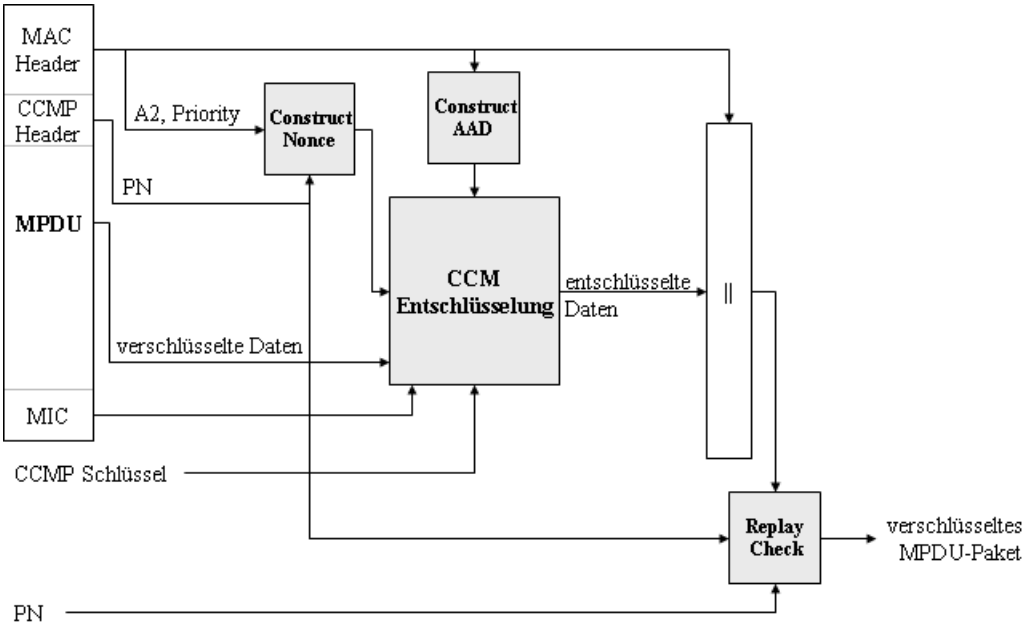

Abbildung 2.15: Blockdiagramm der CCMP-Entschlüsselung (vgl. IEEE 2003, S. 60)

2.5 Chipkarten

In den 70er Jahren ermöglichte der rasante Fortschritt der Mikroelektronik, Datenspeicher und Rechnerlogik auf einem einzigen kleinen Siliziumplättchen zu integrieren. Im Jahre 1968 wurde die Idee, einen Schaltkreis in eine Identifikationskarte zu integrieren, von den Erfindern Jürgen Dethloff und Helmut Grötrupp in Deutschland zum Patent angemeldet. Es folgte 1970 eine ähnliche Anmeldung in Japan von Kunitaka. In Bewegung kam die Entwicklung der Chipkarten aber erst, nachdem in Frankreich Roland Moreno seine Chipkartenpatente anmeldete. Nun war die Halbleiterindustrie in der Lage, die erforderlichen integrierten Schaltungen zu akzeptablen Preisen zu liefern. Auf Grund technologischer Probleme war der Weg vom Prototyp zum marktgerechten Produkt aber noch lang.

Im Jahre 1984 führte ein erfolgreicher Feldversuch mit Telefonkarten von der französischen *Post, Telegraph and Telephone Administration* (PTT) zu einem großen Durchbruch. Bei diesem Feldversuch erfüllten die Chipkarten die gestellten Erwartungen wie Manipulationssicherheit und Zuverlässigkeit. In Deutschland fand 1984/85 ein Pilotversuch mit verschiedenen Kartentechnologien statt. Dabei handelte es sich um Karten mit Magnetstreifen, Karten mit optischer Speicherung und Chipkarten. Die Chipkarte ging dabei als Sieger hervor. Die Chipkarten hatten nicht nur eine hohe Manipulationssicherheit und Zuverlässigkeit, sondern versprachen auch für die Zukunft eine größere Flexibilität in der Anwendung. Nach den Pilotprojekten war der Siegeszug der Chipkarten unaufhaltsam. In Frankreich waren 1986 bereits

mehrere Millionen Chipkarten zum Telefonieren im Umlauf. Die Anzahl stieg bis zum Jahr 1997 auf mehrere hundert Millionen weltweit.

Die Telefonkarten waren kostengünstige Speicherchips mit einer speziellen Sicherheitslogik. Komplexere und größere Mikroprozessorchips wurden ebenfalls zuerst in der Telekommunikation, und zwar dem Mobilfunk eingesetzt. Die Deutsche Bundespost führte 1988 eine moderne Mikroprozessorkarte als Berechtigungskarte für das analoge Mobilfunknetz (C-Netz) ein. Die Chipkarten wurden ins digitale GSM-Netz eingeführt, was durch die positive Erfahrung im C-Netz beeinflusst wurde. Das GSM-Netz wurde 1991 in Betrieb genommen und hat heute über 600 Millionen Teilnehmer weltweit.

Im Bereich des Bankwesens war die Einführung der Chipkarten nicht ganz so rasant. Diese Chipkarten mussten mit kryptographischen Verfahren ausgestattet werden. Der erste Pilotversuch fand 1982/83 in Frankreich mit 60000 Karten statt. Erst 10 Jahre später waren alle französischen Banken mit Chipkarten ausgerüstet. In Deutschland fand der erste Feldversuch 1984/85 statt, aber erst 1996 gab der zentrale Kreditausschuss (ZKA) eine Spezifikation für die multifunktionale Eurocheque-Karte mit Chip heraus. In Österreich wurden 1996 Chipkarten mit POS-Funktion (*Point of Sell*), elektronischer Geldbörse und möglichen anderen Zusatzanwendungen landesweit herausgegeben. Somit war Österreich weltweit das erste Land mit einem flächendeckenden elektronischen Geldbörsensystem.

In Deutschland hat heute beinahe jeder mindestens eine Chipkarte, da bei der Einführung der Krankenversicherungskarten mit Chip über 70 Millionen Chipkarten an die gesetzlich Versicherten ausgegeben wurden. Heute sind weltweit in vielen Ländern Chipkarten im Gesundheitsbereich im Einsatz (vgl. Rankl und Effing 2002, S. 1-6).

2.5.1 Chipkartenarten

Je nach Verwendungszweck kommen unterschiedliche Chipkarten zum Einsatz. Man kann die Chipkarten in die Gruppen Speicherkarten, Mikroprozessorkarten und kontaktlose Chipkarten unterteilen.

Speicherkarten

Die ersten Speicherkarten waren Telefonkarten. Diese werden im Voraus bezahlt, und der Betrag wird im Chip elektronisch gespeichert. Bei der Anwendung wird der gespeicherte Wert um den verbrauchten Betrag reduziert. Durch eine entsprechende Sicherheitslogik wird verhindert, dass der Wert im Chip überschrieben werden kann. Eine Reduzierung des Betrags ist somit irreversibel. Solche Speicherkarten können überall da eingesetzt werden, wo gegen Vorbezahlung eine Ware bzw. Dienstleistung bargeldlos verkauft werden soll (z. B.

Kantinen, Schwimmbäder, Parkgebühren und vieles andere). Solche Speicherkarten haben eine einfache Technik und sind günstig in der Herstellung. Der Nachteil ist, dass sie nur einmal verwendet werden können und dann entsorgt werden müssen. Speicherkarten können auch als Ausweiskarten verwendet werden. Beispielsweise werden im Chip der Krankenversicherungskarte die Daten gespeichert, die sonst auf dem Krankenschein eingetragen waren.

Speicherkarten sind durch das Fehlen eines Prozessors von der Funktion her beschränkt. Durch einfache Sicherheitslogiken können die Daten vor Manipulationen geschützt werden. Sie eignen sich besonders für Wertkarten und Ausweiskarten, bei denen es auf den günstigen Herstellungspreis ankommt (vgl. Rankl und Effing 2002, S. 7).

Mikroprozessorkarten

Die ersten Mikroprozessorkarten waren die Bankkarten in Frankreich. Diese konnten geheime Schlüssel sicher speichern und Kryptoalgorithmen ausführen. Das ermöglichte die Realisierung von Offline-Zahlungssystemen mit hohem Sicherheitsniveau. Der Mikroprozessor in der Chipkarte ist frei programmierbar, womit die Funktionalität von Mikroprozessorkarten nur durch den verfügbaren Speicher und die Rechenleistung beschränkt ist. Mikroprozessorkarten werden auch als *Smart Cards* bezeichnet.

Durch die Massenproduktion von Mikroprozessorkarten sind die Herstellungskosten stark gefallen. Und es wurden neue Anwendungsgebiete erschlossen. Insbesondere durch die Anwendung in Mobiltelefonen kam es zur internationalen Verbreitung der Chipkarte. Mikroprozessorkarten werden auch zur Zugangskontrolle, als geschützte Datenspeicher, zur Signierung von Dokumenten und als elektronische Geldbörse eingesetzt. Weiterhin gibt es multifunktionale Chipkarten, die mehrere Anwendungen im Chip enthalten.

Moderne Mikroprozessorkarten sind mit Betriebssystemen ausgestattet, die ein nachträgliches Laden neuer Anwendungen auf die Chipkarte erlauben, ohne die Sicherheit der bestehenden Anwendungen zu gefährden. Dies ermöglicht den flexiblen Einsatz von Mikroprozessorkarten in vielen kleinen Anwendungsgebieten.

Die Vorzüge der Mikroprozessorkarte sind die hohe Speicherkapazität, die sichere Speicherung geheimer Daten und die Fähigkeit, Kryptoalgorithmen berechnen zu können. Das Potenzial der Chipkarte ist noch nicht erschöpft und wird durch den Fortschritt der Halbleitertechnologie stetig erweitert (vgl. Rankl und Effing 2002, S. 7-9).

Kontaktlose Chipkarten

In den letzten Jahren wurden kontaktlose Chipkarten zur Produktionsreife gebracht. Die Energie- und Datenübertragung erfolgt ohne galvanische Kopplung zwischen Chipkarte und Terminal. Heute sind sowohl Speicher- als auch Mikroprozessorkarten verfügbar. Durch die höhere Leistungsaufnahme funktioniert die kontaktlose Mikroprozessorkarte meist nur in einer Entfernung von wenigen Zentimetern zum Terminal. Kontaktlose Speicherkarten können in einer Entfernung von bis zu einem Meter erreicht werden. Solche Speicherkarten müssen nicht in die Hand genommen werden, damit sie funktionieren, somit sind sie besonders für den Einsatz zur Zugangskontrolle, im öffentlichen Personennahverkehr, als Skipass oder als Flugticket geeignet.

Nicht geeignet sind diese Chipkarten als elektronische Geldbörse. Hier ist eine Willenserklärung des Karteninhabers notwendig, sonst könnte ein Betrüger einfach unbemerkt Geld von der Chipkarte abheben. Eine Lösung hierfür ist die so genannten Dual-Interface-Karte (auch Combicard genannt). Diese verfügt sowohl über eine kontaktbehaftete als auch über eine kontaktlose Schnittstelle (vgl. Rankl und Effing 2002, S. 9-10).

2.5.2 PIN-Verfahren

Die häufigste Art der Benutzeridentifikation für Chipkarten ist die Prüfung der PIN (*personal identification number*). Diese wird auch manchmal als CHV (*card holder verification*) bezeichnet.

Die PIN besteht normalerweise aus vier Stellen, die sich aus den Dezimalzahlen „0“ bis „9“ zusammensetzen. Die PIN ist meist auf vier Stellen beschränkt, weil ein Mensch sich diese merken muss. Der Grund für eine reine Zahlenkonstellation ist die numerische Tastatur von Terminals. Die PIN wird auf der Tastatur des Terminals oder Computers eingegeben und zur Chipkarte gesendet. Diese vergleicht den Wert mit dem gespeicherten Referenzwert und sendet das Ergebnis an das Terminal. Der Standard ISO/IEC 7814-4 beschreibt dafür das entsprechende Chipkartenkommando “*VERIFY*“ (vgl. Rankl und Effing 2002, S. 503-509).

Es gibt statische und veränderbare PINs. Eine statische PIN ist nicht mehr veränderbar. Sollte diese bekannt werden, dann muss der Besitzer der Karte die Chipkarte konsequenterweise zerstören und eine neue Chipkarte besorgen. Chipkarten mit einer veränderbaren PIN ermöglichen dem Benutzer, seine eigene PIN zu wählen. In diesem Fall besteht die Gefahr, dass die Karteninhaber triviale PINs, wie „1234“, „4711“ und „0815“ wählen. Eine Prüfung auf solche Trivialwerte findet meist nicht auf den Chipkarten statt. Zum Ändern der PIN wird immer die aktuelle PIN verlangt und mit übergeben.

Es gibt Anwendungen, die Transport-PINs verwenden. Dabei wird auf der Chipkarte eine zufällige PIN gespeichert, die der Kartenbenutzer in einem PIN-Brief erhält. Der Benutzer muss dann bei der ersten Eingabe die PIN ändern. Ähnlich verhält es sich beim „Nullpin-Verfahren“. Hier wird auf der Chipkarte eine Trivial-PIN, wie „0000“ gespeichert, die wiederum vom Benutzer geändert werden muss (vgl. Rankl und Effing 2002, S. 503-509).

2.5.3 Authentifizierung

Aktuelle Betriebssysteme für Mikroprozessorkarten bieten die Möglichkeit, das Terminal und die Chipkarte gegenseitig zu authentifizieren. Der Standard ISO/IEC 7814-4 beschreibt dafür verschiedene Chipkartenkommandos.

Das Authentifizierungsverfahren beruht auf der Kenntnis eines gemeinsamen Geheimnisses. Dieses Geheimnis ist ein kartenindividueller symmetrischer Schlüssel und sollte von einem einzigartigen Merkmal der Chipkarte abhängig sein. Dafür eignet sich die Seriennummer der Chipkarte, die in dieser Arbeit als Chipkartennummer bezeichnet wird. Die Chipkartennummer kann durch ein Kommando, wie „*GET CHIP NUMBER*“ abgefragt werden.

Zur Überprüfung der Authentizität wird das Challenge-Response-Verfahren eingesetzt. Das Kommando „*INTERNAL AUTHENTICATE*“ gibt dem Terminal die Möglichkeit, die Authentizität der Chipkarte zu prüfen. Das Terminal ermittelt den Schlüssel K der Chipkarte mit Hilfe der Chipkartennummer ID_C. Dann generiert das Terminal eine Zufallszahl N_T als *Challenge*, die als Parameter des Kommandos mit an die Chipkarte übergeben wird. Die Chipkarte verschlüsselt N_T mit dem symmetrischen Schlüssel K und sendet das Ergebnis $\{N_T\}_K$ als *Response* zurück an das Terminal. Das Terminal entschlüsselt das Ergebnis und prüft, ob der entschlüsselte Wert dem generierten N_T entspricht. Die Abbildung 2.16 zeigt ein Beispiel für einen Kommandoablauf einer solchen Authentifizierung (vgl. Rankl und Effing 2002, S. 465-467).

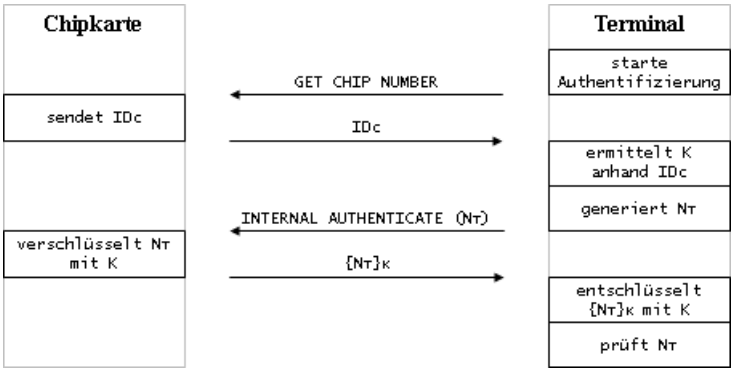

Abbildung 2.16: Beispiel für einen Kommandoablauf mit „*INTERNAL AUTHENTICATE*“ (vgl. Rankl und Effing 2002, S. 467)

Zum Beweis der Authentizität des Terminals gegenüber der Chipkarte gibt es das Kommando „*EXTERNAL AUTHENTICATE*“. Hierfür muss das Terminal zuvor nicht nur den Schlüssel K mittels der Chipkartennummer ID_C ermitteln, sondern auch ein *Challenge* von der Chipkarte anfordern. Dafür legt der Standard ISO/IEC 7816-4 das Kommando „*GET CHALLENGE*“ fest. Hier generiert die Chipkarte eine Zufallszahl N_C und sendet sie als Antwort des Kommandos zurück. Das Terminal muss N_C mit K verschlüsseln und sendet das Ergebnis $\{N_C\}_K$ als Parameter des Kommandos „*EXTERNAL AUTHENTICATE*“ an die Chipkarte. Diese entschlüsselt den Wert und überprüft das Ergebnis mit der zuvor generierten Zufallszahl. Anschließend antwortet die Chipkarte dem Terminal mit einer Nachricht, die das Ergebnis der Authentifizierung enthält. Wenn die Authentifizierung erfolgreich war, dann gibt die Chipkarte entsprechende Daten bzw. Funktionen auf der Chipkarte frei. Die Abbildung 2.17 zeigt den beschriebenen Kommandoablauf (vgl. Rankl und Effing 2002, S. 467-468).

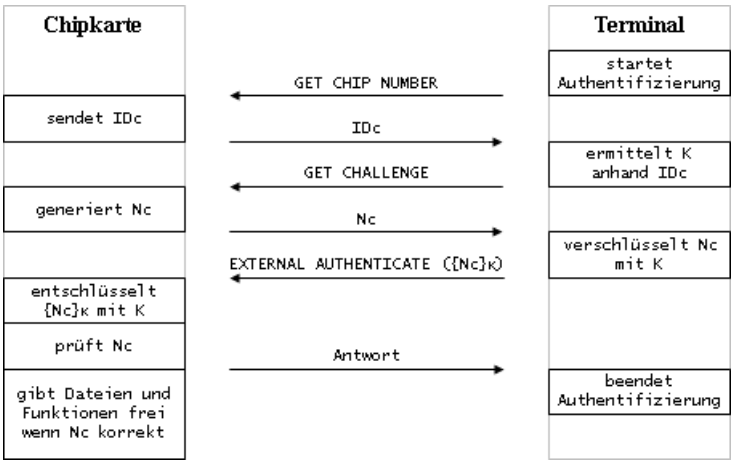

Abbildung 2.17: Beispiel für einen Kommandoablauf mit „*EXTERNAL AUTHENTICATE*“ (vgl. Rankl und Effing 2002, S. 468)

Führt man „*INTERNAL AUTHENTICATE*“ und „*EXTERNAL AUTHENTICATE*“ nacheinander aus, dann haben sich Chipkarte und Terminal gegenseitig authentifiziert. Mit dem Standard ISO/IEC 7814-8 wurde ein Kommando festgelegt, das beide Authentifizierungsschritte zusammenfasst. Der Name des Kommandos ist „*MUTUAL AUTHENTICATE*“. Durch dieses einzelne Authentifizierungskommando wird die Sicherheit der gegenseitigen Authentifizierung erhöht, da keine unlauteren Kommandos mehr zwischen den zwei Authentifizierungen eingefügt werden können. Zusätzlich ist es nicht mehr möglich, durch das Abhören der Kommunikation Klartext-Schlüssel-Paare zu erhalten.

Bevor das Kommando aufgerufen werden kann, muss das Terminal den Schlüssel K anhand der Chipkartennummer ID_C ermitteln. Die Chipkartennummer erhält das Terminal durch das Kommando „*GET CHIP NUMBER*“. Dann muss das Terminal einen Zufallswert N_C mittels

des Kommandos „*GET CHALLENGE*“ von der Chipkarte abfragen. Anschließend erzeugt das Terminal seine eigene Zufallszahl N_T. Nun verschlüsselt das Terminal N_T, N_C und ID_C mit dem Schlüssel K und sendet das Ergebnis $\{N_T, N_C, ID_C\}_K$ als Parameter des Kommandos „*MUTUAL AUTHENTICATE*“ an die Chipkarte. Diese entschlüsselt mit K und prüft die entschlüsselten Werte ID_C und N_C mit den eigenen Werten. Wenn beide Werte korrekt sind, dann verschlüsselt die Chipkarte die Werte N_C und N_T, wobei die Positionen der beiden Werte vertauscht werden. Das Ergebnis $\{N_C, N_T\}_K$ sendet die Chipkarte als Antwort des Kommandos „*MUTUAL AUTHENTICATE*“ an das Terminal. Das Terminal entschlüsselt die Werte und prüft seinerseits N_T. Wenn der entschlüsselte Wert N_T mit dem generierten übereinstimmt, dann war die gegenseitige Authentifizierung erfolgreich. Die Abbildung 2.18 zeigt beispielhaft den Ablauf einer solchen Authentifizierung (vgl. Rankl und Effing 2002, S. 468-469).

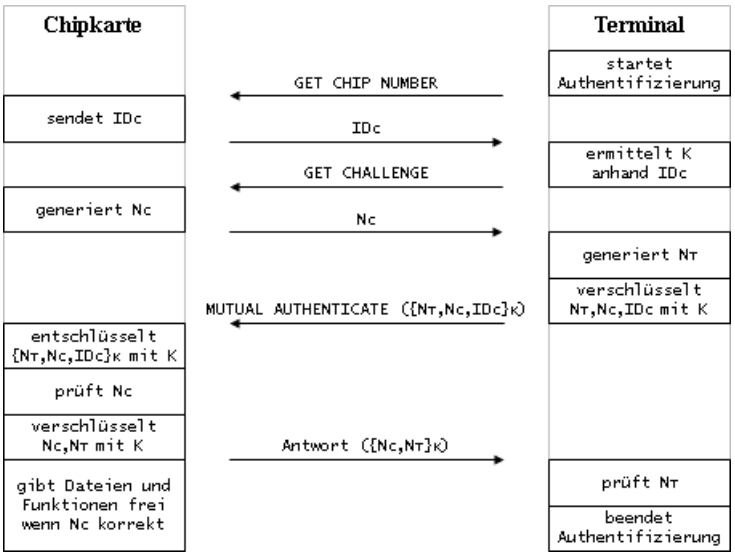

Abbildung 2.18: Beispiel für einen Kommandoablauf mit „*MUTUAL AUTHENTICATE*“ (vgl. Rankl und Effing 2002, S. 469)

2.5.4 Secure Messaging

Nach Rankl und Effing (2002) wurde im Standard ISO/IEC 7816-4 und ergänzend im Standard ISO/IEC 7816-8 ein sehr flexibles, aber dadurch auch aufwendiges und komplexes Verfahren für das *Secure Messaging* definiert.

Die Integrität wird durch eine *cryptographic checksum* (CCS) geschützt. Die CCS wird über die Daten mit einem Blockchiffre-Algorithmus berechnet und entspricht der Bedeutung des MIC. Zum Schutz der Vertraulichkeit können die Daten zusätzlich verschlüsselt werden. Zur

Verschlüsselung wird ebenfalls ein Blockchiffre-Algorithmus eingesetzt. Als Bedingung müssen beide Kommunikationspartner über denselben symmetrischen Schlüssel verfügen.

Ergänzend zur CCS und Verschlüsselung kann ein Sendefolgezähler (*send sequence counter*) eingesetzt werden. Dieser wird von der Chipkarte mit einem Zufallswert initialisiert und das Terminal kann diesen Wert abfragen. Danach wird der Sendefolgezähler für jede Nachricht inkrementiert und in den Daten der Nachrichten so eingebunden, dass der Wert nicht unbemerkt verändert werden kann (vgl. Rankl und Effing 2002, S. 433-441).

2.6 EAP-SIM

Es ist zurzeit ein EAP-Verfahren in der Entwicklung, das bereits Chipkarten für die Authentifizierung verwendet. Dieses Verfahren nennt sich EAP-SIM und wird in H. Haverinen u. a. (2004) spezifiziert. Mit der Spezifikation von EAP-SIM wird ein EAP-Mechanismus festgelegt, der das *GSM Subscriber Identity Module* (SIM) für die Authentifizierung und Schlüsselvereinbarung verwendet. Dabei wird die GSM-Authentifizierung und -Vereinbarung erweitert, indem mehrere Authentifizierung-*Triplets*[5] miteinander kombiniert werden können, um stärkere Authentifizierungsantworten und Sitzungsschlüssel zu erhalten (vgl. H. Haverinen u. a. 2004, S. 1).

Das *Global System for Mobile Communication* (GSM) war weltweit die erste standardisierte Anwendung einer auf Chipkarten basierenden Authentifizierung. Das GSM ist ein Mobilfunkstandard und wurde von 1982 bis 1990 entwickelt. Beim GSM erfolgt die Authentifizierung über die so genannte SIM-Karte, die eine kleine Mikroprozessorkarte ist (vgl. Eckert 2003, S. 633).

Die GSM-Authentifizierung ist ein Challenge-Response-Verfahren. Die SIM-Karte erhält von einem GSM-Server eine 128-Bit-Zufallszahl $RAND$ als *Challenge*. Die SIM-Karte bestimmt dann mit $RAND$ und ihrem individuellen symmetrischen Schlüssel K_I, die 32-Bit-Antwort $SRES$ (*Signed Response*) und einen abgeleiteten 64-Bit-Schlüssel K_C. Die Kombination der zusammenhängenden Werte $RAND$, $SRES$ und K_I wird *Triplet* genannt. Ein GSM-Server bereitet normalerweise mehrere *Triplets* vor, um sein *Authentication Center* (AC) zu entlasten (vgl. Eckert 2003, S. 636).

[5] *Triplet* ist die englische Bezeichnung einer Dreiergruppe. In diesem Zusammenhang besteht die Dreiergruppe aus 3 Werten.

2.6.1 Authentifizierung

Die Abbildung 2.19 zeigt den Ablauf einer vollständigen EAP-SIM-Authentifizierung zwischen einem EAP-Client und einem EAP-Server. Es ist zu beachten, dass die Kommunikation mit EAP-Nachrichten nicht direkt erfolgt, sondern zwischen dem EAP-Client und dem EAP-Server ein Authentifikator als übermittelt fungiert, was beispielsweise in Kapitel 2.4.1 beschrieben wurde.

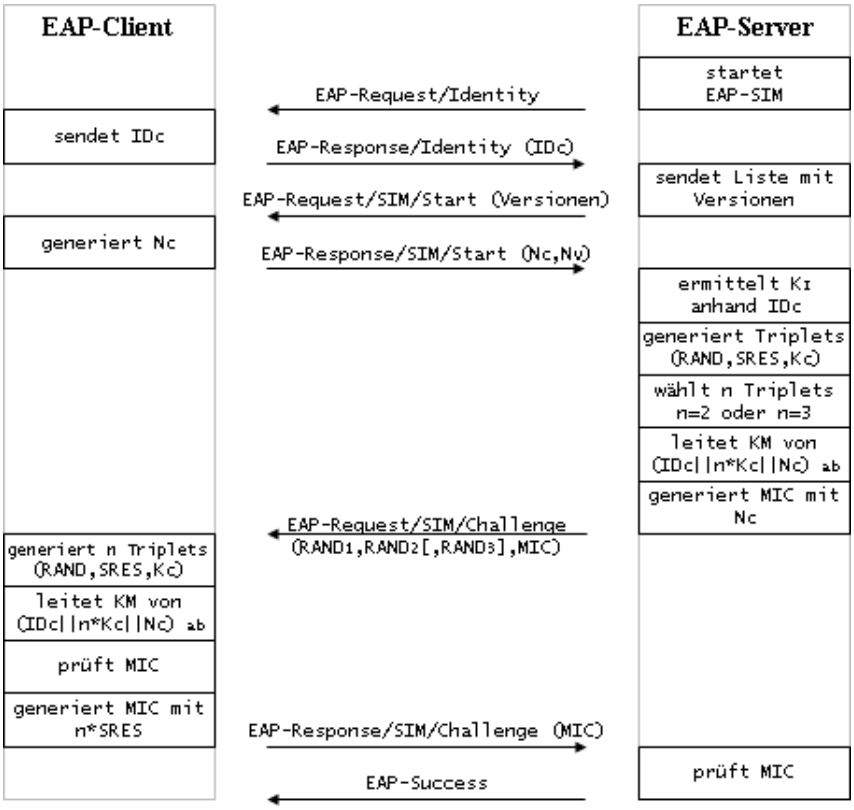

Abbildung 2.19: Ablauf einer vollständigen EAP-SIM-Authentifizierung (vgl. H. Haverinen u. a. 2004, S. 8)

Die erste *EAP-Request*-Nachricht ist vom Typ *Identity*. Die entsprechende Antwort vom EAP-Client beinhaltet die Identität ID_C als *International Mobile Subscriber Identity* (IMSI)[6] oder als Bezeichnung der temporären Identität, die dem EAP-Client zugewiesen wurde.

Nun folgen EAP-Nachrichten von Typ SIM, wobei SIM dem Wert 18 entspricht. Diese Nachrichten werden noch zusätzlich in Subtypen unterteilt. Der erste verwendete Subtyp ist *Start*. Die *EAP-Request*-Nachricht vom Typ *SIM/Start* beinhaltet eine Liste der Versionen von EAP-SIM, die vom EAP-Server unterstützt werden. Daraufhin generiert der EAP-Client eine

[6] Die IMSI ist eine weltweit eindeutige Identifikationsnummer auf der SIM-Karte

Zufallszahl N_C und sendet diese zusammen mit der gewählten EAP-SIM-Versionsnummer N_V als *EAP-Response*-Nachricht vom Typ *SIM/Start* an den EAP-Server.

Als nächstes ermittelt der EAP-Server K_I anhand von ID_C und generiert bis zu 5 *Triplets* $(RAND, SRES, K_C)$ für diesen EAP-Client. Der EAP-Server nimmt 2 oder 3 der *Triplets* und leitet den symmetrischen Schlüssel MK, von der Kombination aus ID_C, den K_C-Werten und N_C, ab. Jetzt sendet der EAP-Server eine *EAP-Request*-Nachricht vom Typ *SIM/Challenge*, wobei *Challenge* ein weiterer Subtyp ist. Die Nachricht beinhaltet alle $RAND$-Werte der verwendeten *Triplets*, und einen 128-Bit-MIC. Dieser MIC wird über die Kombination aus der ganzen EAP-Nachricht und dem Wert N_C unter Verwendung des MK gebildet. Sobald der EAP-Client diese Nachricht empfangen hat, kann dieser mittels der $RAND$-Werte und seines Schlüssels K_I die *Triplets* $(RAND, SRES, K_C)$ vervollständigen und KM ableiten. Anschließend bestimmt der EAP-Client selbst den MIC' und vergleicht diesen mit dem empfangenen MIC. Wenn die empfangene MIC korrekt ist, dann hat sich der EAP-Server gegenüber dem EAP-Client erfolgreich authentifiziert, weil N_C in die Berechnung mit einfloss. Nun bestimmt der EAP-Client den MIC über die Kombination aus *EAP-Response*-Nachricht vom Typ *SIM/Start* und allen $SRES$-Werten aus den ermittelten *Triplets*. Diesen MIC sendet der EAP-Client mit dieser *EAP-Response*-Nachricht an den EAP-Server. Nachdem der EAP-Server die Antwort erhalten hat, bestimmt dieser den MIC' über die Nachricht und den $SRES$-Werten und vergleicht diesen mit dem empfangenden MIC. Wenn beide Werte gleich sind, dann hat sich der EAP-Client gegenüber dem EAP-Server authentifiziert, was der EAP-Server durch eine *EAP-Success*-Nachricht bestätigt (vgl. H. Haverinen u. a. 2004).

2.6.2 Sicherheitsmerkmale

In Eckert (2003) werden einige Sicherheitsprobleme von GSM aufgezeigt, die vom EAP-SIM behoben werden.

Sitzungsschlüssel GSM verwendet nur 64-Bit-Sitzungsschlüssel. EAP-SIM leitet dagegen seinen eigenen Sitzungsschlüssel von zwei bis drei *Triplets* und einem Zufallswert vom EAP-Client ab.

Integrität GSM hat für die Integrität keinen Schutz. EAP-SIM ermittelt einen 128-bit großen MIC für die Nachrichten, sobald er den Sitzungsschlüssel ermittelt hat.

Authentifizierung GSM führt eine einseitige Authentifizierung durch, wobei sich die SIM-Karte gegenüber dem GSM-Server authentifiziert. EAP-SIM führt dagegen eine beidseitige Authentifizierung durch, dabei werden die *Response*-Werte nicht direkt übergeben, sondern bei der Berechnung des MICs mit einbezogen. Da die *Response*-

Werte nicht mit übermittelt werden, kann der Empfänger nicht mit hundertprozentiger Sicherheit sagen, dass der Kommunikationspartner authentisch ist. Die Höhe der Wahrscheinlichkeit, dass der Kommunikationspartner authentisch ist, hängt vom verwendeten Algorithmus für die MIC-Bestimmung ab.

3 Analyse eines Beispielszenarios

Die Sicherheitsanforderungen für die zu entwerfende Sicherheitsarchitektur werden anhand eines Beispielszenarios ermittelt. Zunächst wird das Szenario beschrieben und die allgemeinen Anforderungen formuliert. Nachfolgend wird anhand dieser Informationen eine Risikoanalyse durchgeführt. Die Sicherheitsstrategie legt dann die Sicherheitsanforderungen fest, die von der Risikoanalyse abgeleitet werden.

3.1 Szenario

In diesem Szenario geht es um ein fiktives Unternehmen, das ein kabelloses lokales Netzwerk für die Mitarbeiter zur Verfügung stellen möchte. Dafür soll ein bereits eingesetztes LAN durch ein innerbetriebliches und flächendeckendes Wireless-LAN (siehe Kapitel 2.2) ergänzt werden. Die Sicherheit des Wireless-LANs soll durch den Einsatz von Chipkarten (siehe Kapitel 2.5) gewährleistet werden.

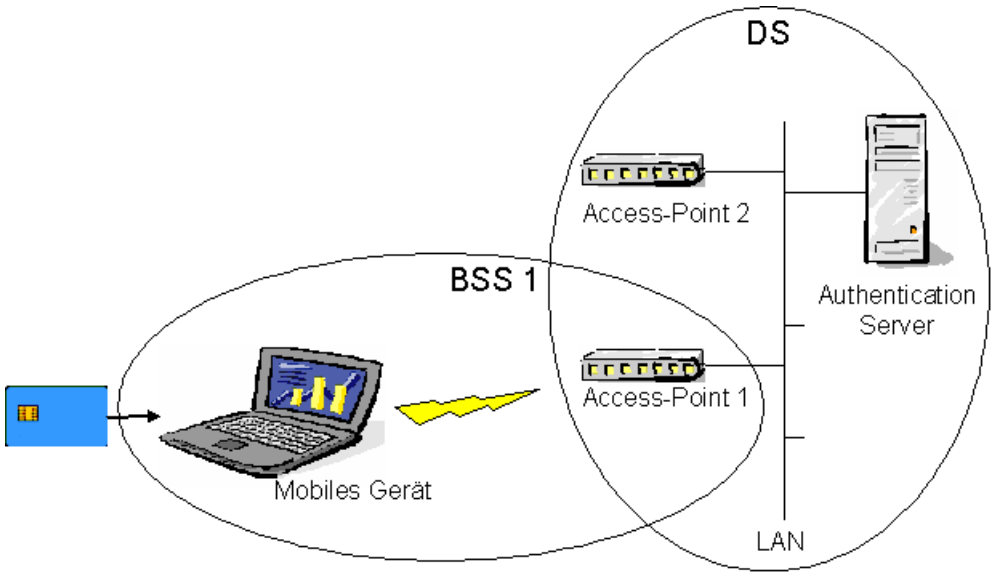

Abbildung 3.1: Grobarchitektur WLAN mit Chipkarte

Als Chipkarten stehen Mikroprozessorkarten (siehe Kapitel 2.5.1) zur Verfügung, die bisher zum Öffnen von Türen und zum Signieren von Dokumenten eingesetzt werden. Jeder Mitarbeiter hat eine dieser Chipkarten. Die mobilen Geräte der Mitarbeiter werden mit Hardware ausgestattet, die den Zugriff auf diese Chipkarten und das Wireless-LAN erlaubt.

Für eine flächendeckende Realisierung ist das Wireless-LAN als ESS-Netzwerk (siehe Kapitel 2.2) zu betreiben. Die Access-Points sind an das LAN anzuschließen, wodurch das LAN als *Distribution System* agiert. Zur Zentralisierung der Authentifizierungen ist zusätzlich ein Authentication-Server in das LAN zu integrieren. Daraus ergibt sich eine Architektur wie in Abbildung 3.1 dargestellt.

3.2 Generelle Anforderungen

Die zu entwickelnde Sicherheitsarchitektur muss für das Wireless-LAN die Schutzziele (siehe Kapitel 2.1) Authentizität, Datenintegrität, Informationsvertraulichkeit und Verfügbarkeit gewährleisten. Einem Benutzer ist der Zugang zum Netzwerk nur erlaubt, wenn die Authentizität des mobilen Geräts, der Chipkarte und des Benutzers bewiesen ist. Zur Überprüfung der Authentizität des Benutzers steht das PIN-Verfahren der Chipkarte (siehe Kapitel 2.5.2) zur Verfügung. Jede Chipkarte hat eine eindeutige Chipkartennummer, die zur Identifikation zu verwenden ist. Das mobile Gerät ist durch die hardwaretechnische Vorraussetzung für Wireless-LAN im Besitz einer MAC-Adresse, die über alle Netzwerkgeräte einmalig ist. Somit ist das mobile Gerät über diese MAC-Adresse zu identifizieren. Die Kommunikation zwischen dem mobilen Gerät und dem Access-Point ist zu verschlüsseln.

3.3 Risikoanalyse

Bei der Risikoanalyse werden die Risiken der Bedrohungen für das System bewertet. Zunächst werden die Bedrohungen durch Bedrohungsbäume dargestellt. Die Wurzel eines Bedrohungsbaums definiert ein Angriffsziel, was eine mögliche Bedrohung für das System darstellt. Abgehend vom Angriffziel, werden die Zwischenziele definiert, die zum Erreichen des Angriffziels beitragen. Diese Zwischenziele müssen entweder gemeinsam erreicht werden, oder jedes Zwischenziel ist eine Alternative zum Erreichen des Angriffsziels. Im Baum werden nur UND-Knoten mit „UND“ gekennzeichnet, alle nicht gekennzeichneten Knoten sind ODER-Knoten. Jedes Zwischenziel kann weitere untergeordnete Zwischenziele besitzen. Jedes Blatt des Baums entspricht einem Angriffsschritt und jeder mögliche Pfad von den Blättern zu der Wurzel beschreibt einen möglichen Weg zum Erreichen des globalen Angriffziels. Zur Bewertung wird die Wahrscheinlichkeit für das Eintreten des Angriffsziels ermittelt

und der mögliche Schaden, den es verursacht. Zur Bestimmung der Wahrscheinlichkeit des Angriffsziels wird zunächst die Eintrittswahrscheinlichkeit eines jeden Angriffschritts bewertet und dann wird die Wahrscheinlichkeit alle Zwischenziele ermittelt. Dabei wird bei einem UND-Knoten die niedrigste Wahrscheinlichkeit der Folgeziele bzw. -blätter übernommen, und bei einem ODER-Knoten die höchste Wahrscheinlichkeit. Der verursachte Schaden wird nur jeweils für das Angriffziel eines Bedrohungsbaums bestimmt und bleibt somit für jeden möglichen Pfad gleich (Eckert 2003). Es kann ein geringer, mittlerer oder hoher Schaden bei einer geringen, mittleren oder hohen Wahrscheinlichkeit verursacht werden.

Die Angriffsziele werden aus den Schutzzielen der generellen Anforderungen abgeleitet. Ein Angriff auf die Authentizität wird durch einen unberechtigten Netzwerkzugang erreicht. Das Abhören der Kommunikationsdaten im Wireless-LAN führt zum Bruch der Informationsvertraulichkeit, das Verändern der Kommunikationsdaten im Wireless-LAN verletzt die Integrität, und das Verursachen einer Überlastung der Kommunikation führt zum Verlust der Verfügbarkeit.

3.3.1 Unberechtigter Netzwerkzugang

Bei einem unberechtigten Netzwerkzugang versucht sich der Angreifer gegenüber dem Netzwerk als ein berechtigter Benutzer auszugeben. Bei einem erfolgreichen Angriff erhält der Angreifer Zugang zum Netzwerk mit allen Rechten und Möglichkeiten, die auch der berechtigte Benutzer hat. Das entspricht einem hohen Schaden für das System. Zum Erreichen des Ziels benötigt der Angreifer ein authentifizierbares mobiles Gerät und eine authentifizierbare Chipkarte. Der Bedrohungsbaum ist in Abbildung 3.2 dargestellt.

In den Besitz eines authentifizierbaren mobilen Geräts kommt der Angreifer, indem er entweder ein mobiles Gerät von einem Mitarbeiter stiehlt oder die Identität eines mobilen Geräts fälscht. Damit der Angreifer ein mobiles Gerät stehlen kann, muss dieses zunächst unbeaufsichtigt sein. Ein Diebstahl wird wohl mit einer mittleren Wahrscheinlichkeit auftreten, da der Mitarbeiter sein mobiles Gerät nicht 24 Stunden lang bei sich hat. Zum Fälschen der Identität benötigt der Angreifer eine gültige MAC-Adresse von einem mobilen Gerät. Da die MAC-Adresse kein Geheimnis ist, wird der Angreifer diese mit hoher Wahrscheinlichkeit erhalten. Zusätzlich benötigt der Angreifer den Kommunikationsschlüssel. Der Schlüssel kann vom Angreifer kompromittiert werden, indem er die Kommunikationsdaten analysiert und den Schlüssel ermittelt. Eine weitere Möglichkeit, um an den Schlüssel zu kommen, ist der Diebstahl eines mobilen Geräts, das den Schlüssel gespeichert hat. Der Aufwand für das Kompromittieren des Schlüssels ist abhängig vom Verschlüsselungsverfahren, und wird somit nur mit geringer Wahrscheinlichkeit durchgeführt. Das Stehlen eines mobilen Geräts wird mit mittlerer Wahrscheinlichkeit durchgeführt, wie bereits zuvor eingeschätzt.

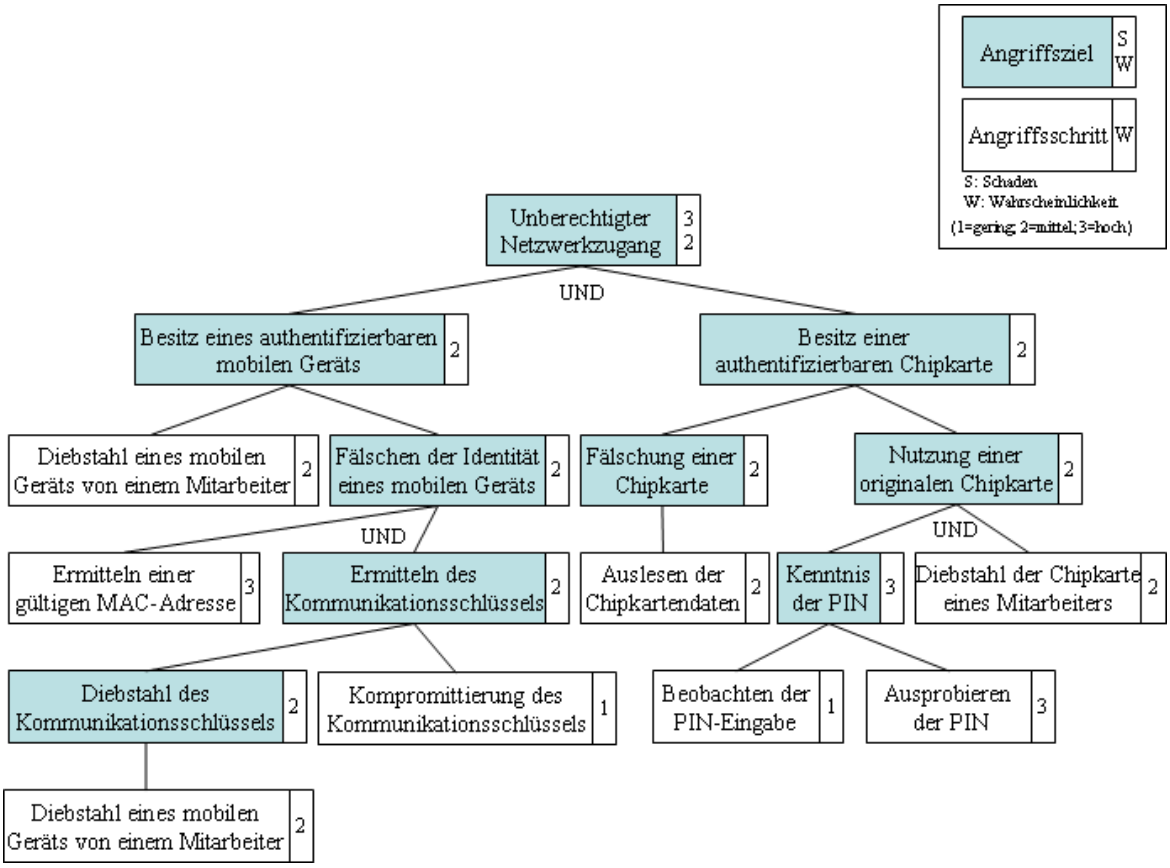

Abbildung 3.2: Angriffsziel: Unberechtigter Netzwerkzugang

In den Besitz einer authentifizierbaren Chipkarte gelangt ein Angreifer durch das Fälschen einer Chipkarte oder durch das Nutzen einer originalen Chipkarte. Zum Fälschen einer Chipkarte muss der Angreifer zunächst diese Chipkarte auslesen, um die Funktion zu verstehen oder eine genaue Kopie erstellen zu können. Die einzige Hürde für den Angreifer ist die Inbesitznahme der originalen Chipkarte, wodurch der Angriff mit einer mittleren Wahrscheinlichkeit auftritt. Wenn der Angreifer eine originale Chipkarte verwenden will, dann muss er diese stehlen und die PIN in Erfahrung bringen. Der Diebstahl wird mit mittlerer Wahrscheinlichkeit auftreten, weil einerseits der Besitzer diese nicht unachtsam legen lassen wird, aber andererseits durch ihre Handlichkeit doch gestohlen werden kann. Die PIN bekommt der Angreifer durch das Beobachten der Eingabe oder durch das Ausprobieren. Der Angreifer wird nur mit einer geringen Wahrscheinlichkeit die PIN-Eingabe beobachten, wenn der Chipkarten-Besitzer aufmerksam ist. Mit hoher Wahrscheinlichkeit wird der Angreifer versuchen die PIN durch einfaches Ausprobieren zu ermitteln, wenn es nicht allzu viele Variationen für die PIN gibt.

3.3.2 Abhören der Kommunikationsdaten

Ein Angreifer versucht die Kommunikation abzuhören, um an geheime Informationen zu gelangen. Wenn der Angriff erfolgreich ist, dann entsteht ein hoher Schaden für das System, weil die geheimen Informationen nicht mehr geheim sind. Abbildung 3.3 zeigt den Bedrohungsbaum für diesen Angriff.

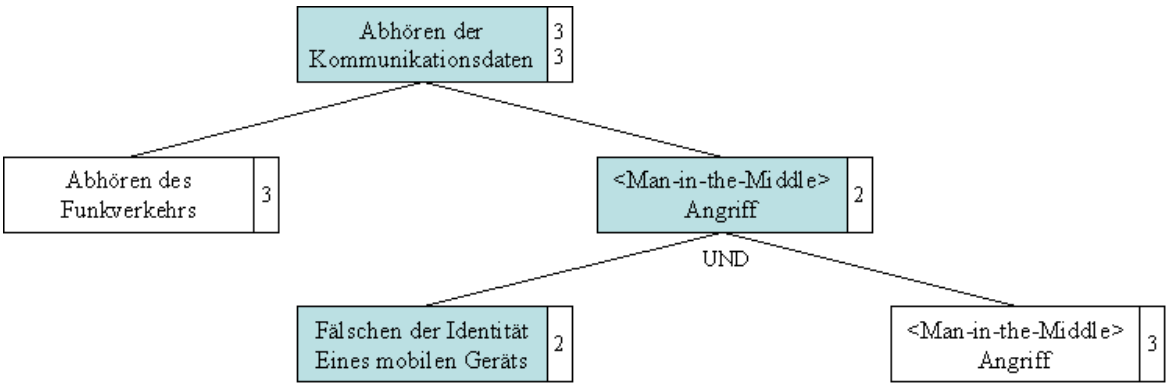

Abbildung 3.3: Angriffsziel: Abhören der Kommunikationsdaten

Ein Angreifer kann das Abhören auf zwei Wege erreichen. Zunächst kann er den Funkverkehr des Wireless-LAN abhören. Dies kann mit hoher Wahrscheinlichkeit auftreten, da die Funksignale überall in Reichweite empfangen werden können. Die zweite Möglichkeit ist ein *Man-in-the-Middle*-Angriff, hier gibt sich der Angreifer gegenüber dem mobilen Gerät als Access-Point aus und gegenüber dem Access-Point als mobiles Gerät. Dafür muss der Angreifer der Identität des mobilen Geräts und des Access-Points fälschen. Das Fälschen der Identität des mobilen Geräts verläuft äquivalent zum „Fälschen der Identität eines mobilen Geräts“ aus dem Kapitel 3.3.1. Daraus folgt, dass der Angriff mit einer mittleren Wahrscheinlichkeit erfolgen wird. Das Fälschen der Identität des Access-Points ist umso einfacher, da dieser seine Identität nicht beweisen muss. Somit wird dies mit einer hohen Wahrscheinlichkeit eintreten.

3.3.3 Verändern der Kommunikationsdaten

Wenn das Verändern der Kommunikationsdaten durch einen Angriff erfolgreich ist, dann entsteht hoher Schaden im System, weil der Wahrheitsgehalt der Daten nicht mehr gewährleistet werden kann.

Ein Angreifer wird versuchen, durch einen *Man-in-the-Middle*-Angriff die Daten zu manipulieren, indem er die Daten vom mobilen Gerät empfängt, diese ändert und dann weiter zum

Access-Point sendet. Dieser Angriff ist gleich dem *Man-in-the-Middle*-Angriff aus dem Kapitel 3.3.2, die mit mittlerer Wahrscheinlichkeit durchgeführt wird.

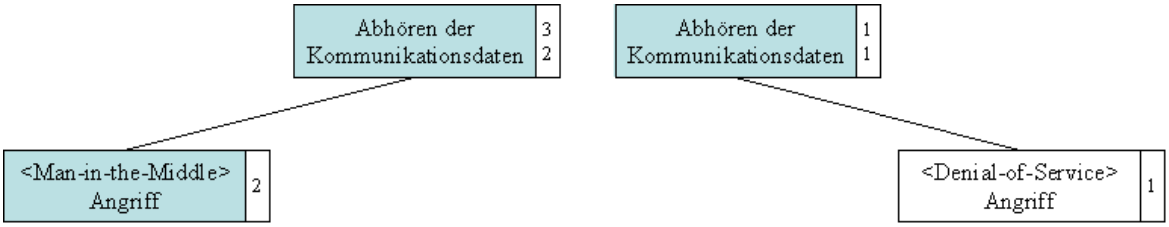

Abbildung 3.4: Angriffsziele: Verändern der Kommunikationsdaten, Verursachen einer Überlastung der Kommunikation

3.3.4 Verursachen einer Überlastung der Kommunikation

Durch eine Überlastung der Kommunikation versucht ein Angreifer, anderen die Möglichkeit des Zugriffs auf das Wireless-LAN zu nehmen. Dies bedeutet einen hohen Schaden für das System.

Um eine Überlastung zu verursachen, wird ein *Denial-of-Service*-Angriff durchgeführt. Bei diesem Angriff werden andauernd Anfragen in möglichst kurzen Abständen gesendet, so dass die Abarbeitung langsamer ist als die Zeitabstände. Irgendwann bricht das System durch die vielen nicht abgearbeiteten Anfragen zusammen. Im internen Wireless-LAN wird dieser Angriff nur mit geringer Wahrscheinlichkeit durchgeführt, da er für den Angreifer keinen Gewinn bringt, sondern nur eine zerstörerische Wirkung auf das Wireless-LAN hat.

3.4 Sicherheitsstrategie

Aus den gewonnenen Erkenntnissen können folgende Sicherheitsanforderungen für das System abgeleitet werden, um die Wahrscheinlicht der Angriffsschritte zu minimieren oder ganz auszuschließen:

- Auf den mobilen Geräten und auf den Chipkarten sind keine Schlüssel zu speichern, die allgemeingültig über alle Geräte bzw. Chipkarten sind und für die Sicherheitsstrategie notwendig sind. Das minimiert den Nutzen sowohl des Diebstahls von mobilen Geräten als auch den Nutzen des Auslesens von Chipkartendaten.

- Die Kommunikation zwischen dem mobilen Gerät und dem Access-Point ist zu verschlüsseln und der Kommunikationsschlüssel soll bei jeder Verbindung mit einem zufälligen Wert neu festgelegt werden. Dadurch hat der Angreifer nur während der jeweiligen Verbindung Zeit den Kommunikationsschlüssel zu kompromittieren, der wiederum nach der Verbindung ungültig wird. Somit wird der Nutzen einer Kompromittierung des Kommunikationsschlüssels sehr gering.
- Das mobile Gerät und der Authentication-Server müssen jeweils die Authentizität des Anderen prüfen. Es ist ein Protokoll zu verwenden, das mit einem symmetrischen Schlüssel (*shared key*) arbeitet. Das führt dazu, dass der Angreifer für das Fälschen der Identität des mobilen Geräts zusätzlich den symmetrischen Schlüssel in Erfahrung bringen muss. Gleichzeitig wird es für den Angreifer schwieriger, sich gegenüber einem mobilen Gerät als Access-Point auszugeben, da ein Access-Point über LAN mit dem Authentication-Server verbunden ist.
- Es darf nicht möglich sein, sicherheitsrelevante Informationen aus Chipkarten direkt auszulesen. Das verringert die Wahrscheinlichkeit, dass Chipkarten gefälscht werden.
- Zwischen der Chipkarte und dem Authentication-Server muss eine gegenseitige Authentifizierung unter Verwendung eines symmetrischen Schlüssels erfolgen. Dadurch wird das Speichern des Schlüssels der Chipkarte auf dem mobilen Gerät vermieden und die Chipkarte kann sich der Authentizität des Kommunikationspartners sicher sein.
- Der Authentication-Server hat die Schlüssel für die mobilen Geräte und für die Chipkarten sicher in einer Datenbank abzulegen. Da nur der Authentication-Server für die Authentifizierungen zuständig ist, dürfen die Schlüsselinformationen nur vom Administrator verwaltet und nicht weitergegeben werden. Der Datenbankeintrag für ein mobiles Gerät setzt sich aus seiner MAC-Adresse und dem symmetrischen Schlüssel zusammen. Äquivalent dazu setzt sich der Datenbankeintrag für eine Chipkarte aus ihrer Chipkartennummer und dem jeweiligen Schlüssel zusammen. Der symmetrische Schlüssel darf nur für das jeweilige mobile Gerät bzw. für die jeweilige Chipkarte gelten.
- Den generellen Anforderungen ist noch zu entnehmen, dass sich der Authentication-Server der Authentizität des Benutzers sicher sein muss. Die Authentizität soll durch das PIN-Verfahren der Chipkarte geprüft werden.

4 Entwurf der Sicherheitsarchitektur

Mit der im Kapitel 3.4 festgelegten Sicherheitsstrategie wird nun eine Sicherheitsarchitektur mit einem entsprechenden Sicherheitsprotokoll entworfen. Die folgenden Anforderungen sind in der Sicherheitsstrategie festgelegt worden:

- Die für die Sicherheitsarchitektur notwendigen allgemeingültigen Schlüssel sind nicht auf Chipkarten und mobilen Geräten zu speichern.
- Das mobile Gerät muss die Authentizität des Authentication-Servers prüfen.
- Der Authentication-Server muss die Authentizität des mobilen Geräts prüfen.
- Die Chipkarte muss die Authentizität des Authentication-Servers prüfen.
- Der Authentication-Server muss die Authentizität der Chipkarte prüfen.
- Der Authentication-Server muss die Authentizität des Benutzers durch die PIN-Abfrage der Chipkarte prüfen.
- Jede Chipkarte und jedes mobile Gerät muss einen individuellen Schlüssel besitzen.
- Der Datenbankeintrag für ein mobiles Gerät setzt sich aus der MAC-Adresse und dem symmetrischen Schlüssel zusammen.
- Der Datenbankeintrag für eine Chipkarte setzt sich aus der Chipkartennummer und dem symmetrischen Schlüssel zusammen.
- Der Authentication-Server muss die Datenbankeinträge der mobilen Geräte und der Chipkarten in einer Datenbank sicher verwalten.
- Die Kommunikation ist zu verschlüsseln.
- Für jede neue Verbindung ist ein zufälliger Kommunikationsschlüssel zu generieren.

4.1 Sicherheitsarchitektur

Als Grundlage für die Sicherheitsarchitektur stehen die WEP-Sicherheitsarchitektur (siehe Kapitel 2.3) und die RSN-Sicherheitsarchitektur (siehe Kapitel 2.4) zur Verfügung.

Die WEP-Sicherheitsarchitektur hat einige schwerwiegende Sicherheitsprobleme (siehe Kapitel 2.3.4). Es wird nur eine Authentifizierung des mobilen Geräts gegenüber dem Access-Point durchgeführt. Für die Verschlüsselung wird derselbe Schlüssel verwendet wie für die Authentifizierung. Weiterhin ist die Verschlüsselung der Kommunikation nicht ausreichend. Somit wird durch die WEP-Sicherheitsarchitektur, was den Authentizitätsprüfungen betrifft, nur eine Anforderung erfüllt.

Die RSN-Sicherheitsarchitektur beinhaltet kein eigenes Authentifizierungsverfahren, sondern bietet die Möglichkeit, beliebige Authentifizierungsverfahren zu verwenden. Dies wird durch die Unterstützung von EAP (siehe Kapitel 2.4.1) erreicht. Weiterhin ist in der RSN-Sicherheitsarchitektur ein Schlüsselmanagement integriert, das das Generieren eines Kommunikationsschlüssels für jede Verbindung garantiert (siehe Kapitel 2.4.2). Zur Verschlüsselung der Kommunikation werden die neuen Protokolle CCMP und TKIP eingesetzt (siehe Kapitel 2.4.3). Durch den zufälligen Kommunikationsschlüssel und die verschlüsselte Kommunikation sind bereits zwei Anforderungen erfüllt. Zusätzlich ist es möglich, durch die Wahl eines geeigneten EAP-Authentifizierungsverfahrens weitere Anforderungen zu erfüllen. Damit ist die RSN-Sicherheitsarchitektur als Grundlage für die zu entwerfende Sicherheitsarchitektur besser geeignet als die WEP-Sicherheitsarchitektur.

Die Komponenten der zu entwickelnden Sicherheitsarchitektur sind die Chipkarte, das mobile Gerät, der Access-Point und der Authentication-Server. Das mobile Gerät und der Access-Point sind bereits von der RSN-Sicherheitsarchitektur bekannt. Die Kommunikation zwischen dem mobilen Gerät und dem Access-Point ist bereits durch die RSN-Sicherheitsarchitektur festgelegt. Für die Authentifizierung werden EAP-Nachrichten in EAPOL-Nachrichten gekapselt und übertragen (siehe Kapitel 2.4.1). Das EAPOL-Protokoll wird durch den Standard IEEE 802.1x festgelegt. Nach einer erfolgreichen Authentifizierung wird jede weitere Kommunikation durch CCMP bzw. TKIP verschlüsselt (siehe Kapitel 2.4.3).

Der Access-Point muss als Übermittler der EAP-Nachrichten zwischen dem mobilen Gerät und dem Authentication-Server dienen. Für den Austausch von EAP-Nachrichten zwischen dem Access-Point und dem Authentication-Server kann das RADIUS-Protokoll eingesetzt werden (siehe Kapitel 2.4.1), das ebenfalls die EAP-Nachrichten in seinen Protokoll-Nachrichten kapseln kann.

Für die Authentifizierung selbst ist ein EAP-Authentifizierungsverfahren notwendig, das den Sicherheitsanforderungen genügt. Das bisher einzige spezifizierte EAP-Authentifizierungsverfahren mit Chipkarte ist EAP-SIM (siehe Kapitel 2.6). Dieses Verfahren führt eine gegen-

seitige Authentifizierung zwischen der Chipkarte und dem Authentication-Server durch. Da aber auch die Authentizitäten des Benutzers und des mobilen Geräts geprüft werden sollen, ist dieses Verfahren nicht geeignet.

Um die Anforderungen bestmöglich zu erfüllen, wird in dieser Arbeit ein neues EAP-Authentifizierungsverfahren entworfen, das von mir mit EAP-CHIP bezeichnet wird. Das EAP-CHIP-Verfahren muss sowohl vom mobilen Gerät als auch vom Authentication-Server entsprechend ihrer Rollen unterstützt werden, damit eine EAP-Authentifizierung durchgeführt werden kann.

Auf der Seite des mobilen Geräts muss das EAP-CHIP-Verfahren mit der Chipkarte kommunizieren. Dabei ist die Kommunikation nach den Standards ISO/IEC 7816-4/8 zu sichern (siehe Kapitel 2.5.3 und 2.5.4). Seitens des Authentication-Servers benötigt das EAP-CHIP-Verfahren Zugriff auf die Datenbank (DB), in der die symmetrischen Schlüssel gespeichert sind.

Die Abbildung 4.1 zeigt die entworfene Sicherheitsarchitektur, deren Komponenten, eingesetzte Protokolle bzw. Verfahren und deren Zusammenhänge.

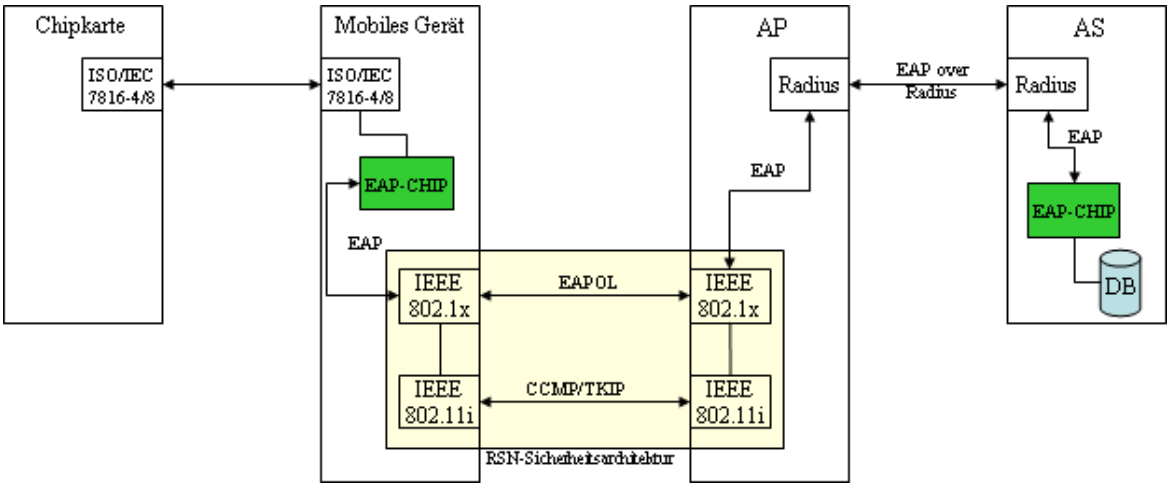

Abbildung 4.1: Sicherheitsarchitektur

4.2 EAP-CHIP-Verfahren

Das neue EAP-Authentifizierungsverfahren EAP-CHIP soll die Sicherheitsanforderungen erfüllen, die für die Authentifizierung relevant sind. Die beteiligten Komponenten sind die Chipkarte C, das mobile Gerät M und der Authentication-Server S. Entsprechend der entworfenen Sicherheitsarchitektur ist es nur dem mobilen Gerät möglich, sowohl mit der Chipkarte als auch mit dem Authentication-Server zu kommunizieren.

Die Chipkarte kennt ihre Chipkartennummer ID_C und ihren symmetrischen Schlüssel K_{CS}. Das mobile Gerät kennt seine MAC-Adresse ID_M und seinen symmetrischen Schlüssel K_{MS}. Der Authentication-Server kann die jeweiligen symmetrischen Schlüssel anhand der Datenbankeinträge (ID_C, K_{CS}) und (ID_M, K_{MS}) aus der Datenbank ermitteln.

Das EAP-CHIP-Verfahren wird nur für eine Neuanmeldung in das Wireless-LAN spezifiziert. Das bedeutet, ein Benutzer möchte sein mobiles Gerät an das Wireless-LAN anmelden und hat dafür seine Chipkarte in den Kartenleser gesteckt.

Nach einer erfolgreichen Authentifizierung muss sowohl das mobile Gerät als auch der Access-Point einen symmetrischen Schlüssel K_{MP} kennen, der nach der Authentifizierung an die RSN-Sicherheitsarchitektur als *Pairwise-Master-Key* übergeben wird (siehe Kapitel 2.4.2). Der Access-Point erhält K_{MP} vom Authentication-Server über das RADIUS-Protokoll, wie in Kapitel 2.4.1 beschrieben.

4.2.1 Authentifizierungsablauf

Nun wird der Ablauf der EAP-CHIP-Authentifizierung festgelegt, wobei gleichzeitig die Nachrichten für das Protokoll festgelegt werden.

Die EAP-Authentifizierung startet mit der Kommunikation zwischen dem mobilen Gerät und dem Authentication-Server. Somit ist es zunächst notwendig, die gegenseitige Authentifizierung zwischen dem mobilen Gerät und dem Authentication-Server durchzuführen.

Die Prüfung der Authentizität wird durch ein Challenge-Response-Verfahren durchgeführt. Dafür sendet das mobile Gerät eine Zufallszahl N_M als *Challenge* an den Authentication-Server. Der Authentication-Server verschlüsselt N_M mit den Schlüssel K_{MS} und sendet das Ergebnis als *Response* zurück an das mobile Gerät. Abschließend prüft das mobile Gerät die *Response* vom Authentication-Server.

Zur Ermittlung des Schlüssels K_{MS} benötigt der Authentication-Server die Identität ID_M vom mobilen Gerät. Mit ID_M kann der Authentication-Server den entsprechenden Datenbankeintrag (ID_M, K_{MS}) finden.
(01) $M \longrightarrow S : ID_M$ [1]

Nun generiert das mobile Gerät einen Zufallswert N_M, der an den Authentication-Server gesendet wird. Dieser verschlüsselt N_M mit dem symmetrischen Schlüssel K_{MS} und sendet das Ergebnis an das mobile Gerät.
(02) $M \longrightarrow S : N_M$
(03) $S \longrightarrow M : \{N_M\}_{K_{MS}}$

Das mobile Gerät entschlüsselt den Wert und vergleicht diesen mit dem selbst generierten Wert N_M. Wenn diese identisch sind, dann ist die Authentizität des Authentication-Servers bewiesen.

Jetzt muss der Authentication-Server die Authentizität des mobilen Geräts prüfen und erstellt dafür eine eigene Zufallszahl N_{S1}, die an das mobile Gerät gesendet wird. Das mobile Gerät verschlüsselt N_{S1} mit K_{MS} und sendet das Ergebnis an den Authentication-Server.
(04) $S \longrightarrow M : N_{S1}$
(05) $M \longrightarrow S : \{N_{S1}\}_{K_{MS}}$

Der Authentication-Server entschlüsselt die empfangene Nachricht und kontrolliert den Wert N_{S1}.

Damit haben sich das mobile Gerät und der Authentication-Server davon überzeugt, dass der jeweils Andere authentisch ist und den Schlüssel K_{MS} kennt.

Als nächstes müssen sich die Chipkarte und der Authentication-Server gegenseitig authentifizieren. Die Chipkarte bietet eine entsprechende Authentifizierungsfunktionalität (siehe Kapitel 2.5.3). Da aber der Authentication-Server nicht direkt mit der Chipkarte verbunden ist, muss das mobile Gerät die notwendigen Nachrichten weiterleiten.

Als erstes benötigt der Authentication-Server die Identität der Chipkarte. Dazu muss das mobile Gerät die Chipkartennummer ID_C von der Chipkarte abfragen und diese mit K_{MS} verschlüsselt an den Authentication-Server senden. Zur Abfrage der Chipkartennummer wird die Chipkartenfunktion „*GET CHIP NUMBER*“ verwendet.
(06) $C \longrightarrow M : ID_C$
(07) $M \longrightarrow S : \{ID_C\}_{K_{MS}}$

Der Authentication-Server kann anhand ID_C aus der Datenbank den Datenbankeintrag ermitteln, der den symmetrischen Schlüssel K_{CS} enthält.

[1] $S \longrightarrow E : N$ bedeutet, dass der Sender S die Nachricht N an den Empfänger E sendet. Eine Nachricht besteht aus einer Liste von Werten, die durch ein Komma getrennt sind. Die Liste muss aus mindestes einem Wert bestehen.

Das mobile Gerät ruft nun die Chipkartenfunktion „*GET CHALLENGE*“ auf, und erhält die Zufallszahl N_C. Das mobile Gerät sendet daraufhin N_C verschlüsselt an den Authentication-Server.
(08) $C \longrightarrow M : N_C$
(09) $M \longrightarrow S : \{N_C\}_{K_{MS}}$

Nun erstellt der Authentication-Server selbst eine neue Zufallszahl N_{S2} und verschlüsselt diese zusammen mit N_C und ID_C mit dem Schlüssel K_{CS}. Der Authentication-Server sendet das Ergebnis an das mobile Gerät. Da der Inhalt der Nachricht gleichzeitig der Parameter für die Chipkartenfunktion „*MUTUAL AUTHENTICATE*“ ist, kann das mobile Gerät diese Funktion direkt aufrufen.
(10) $S \longrightarrow M : \{N_{S2}, N_C, ID_C\}_{K_{CS}}$
(11) $M \longrightarrow C : \{N_{S2}, N_C, ID_C\}_{K_{CS}}$

Die Chipkartenfunktion „*MUTUAL AUTHENTICATE*“ entschlüsselt die Daten mit K_{CS} und überprüft N_C. Wenn N_C korrekt ist, dann ist der Authentication-Server gegenüber der Chipkarte authentifiziert. Jetzt vertauscht die Chipkarte N_C und N_{S2} und verschlüsselt diese mit K_{CS}. Das Ergebnis erhält das mobile Gerät als Antwort, das diese als Nachricht an den Authentication-Server weiterleitet.
(12) $C \longrightarrow M : \{N_C, N_{S2}\}_{K_{CS}}$
(13) $M \longrightarrow S : \{N_C, N_{S2}\}_{K_{CS}}$

Der Authentication-Server entschlüsselt die empfangen Daten mit K_{CS} und überprüft N_{S2}. Wenn N_{S2} korrekt ist, dann ist auch die Chipkarte gegenüber dem Authentication-Server authentifiziert.

Sowohl die Chipkarte als auch der Authentication-Server bestimmen nun unabhängig den Sitzungsschlüssel K_{CM}, der vom Zufallswert N_{S2} abhängt. Der Authentication-Server verschlüsselt K_{CM} und N_M mit K_{MS} und sendet das Ergebnis an das mobile Gerät.
(14) $S \longrightarrow M : \{K_{CM}, N_M\}_{K_{MS}}$

Das mobile Gerät entschlüsselt die Nachricht mit K_{MS} und prüft N_M. Wenn N_M korrekt ist, dann ist K_{CM} aktuell und kann für das *Secure Messaging* eingesetzt werden, wodurch die Kommunikation zwischen der Chipkarte und dem mobilen Gerät abgesichert wird (siehe Kapitel 2.5.4).

Als letztes muss die Authentizität des Benutzers überprüft werden. Dieser gibt die PIN am mobilen Gerät ein. Daraufhin ruft das mobile Gerät die Chipkartenfunktion „*VERIFY*“ mit der PIN als Parameter auf. Die Nachricht wird mit K_{CM} geschützt.
(15) $M \longrightarrow C : \{PIN\}_{K_{CM}}$

Die Chipkarte gibt bei der richtigen PIN die Funktionen der Chipkarte frei (siehe Kapitel 2.5.2). Jetzt muss noch dem Authentication-Server bewiesen werden, dass die richtige PIN eingegeben wurde, ohne das der Authentication-Server die PIN kennen muss.

Da der Authentication-Server bereits von der Authentizität der Chipkarte überzeugt ist, muss die Chipkarte nun den Authentication-Server überzeugen, dass der Benutzer die PIN kennt. Dafür benötigt die Chipkarte erneut den Zufallswert N_{S2} vom Authentication-Server, weil N_{S2} nach dem Aufruf der standardisierten Chipkartenfunktion „*MUTUAL AUTHENTICATE*“ nicht mehr bekannt ist. N_{S2} wird mit K_{CS} verschlüsselt und über das mobile Gerät an die Chipkarte gesendet. Vom mobilen Gerät zur Chipkarte wird die Nachricht durch das *Secure Messaging* zusätzlich mit K_{CM} verschlüsselt.
(16) $S \longrightarrow M : \{N_{S2}\}_{K_{CS}}$
(17) $M \longrightarrow C : \{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}$

Die Chipkarte entschlüsselt die Daten und erstellt eine Antwortnachricht, die sich aus einer Antwort A (z. B. der Text „die PIN ist korrekt“) und dem Wert N_{S2} zusammensetzt und mit K_{CS} verschlüsselt wird. Über die sichere Verbindung erhält das mobile Gerät die Antwortnachricht, die an den Authentication-Server weitergeleitet wird.
(18) $C \longrightarrow M : \{\{A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}$
(19) $M \longrightarrow S : \{A, N_{S2}\}_{K_{CS}}$

Der Authentication-Server entschlüsselt die Nachricht mit K_{CS} und erhält mit der Antwort A die Bestätigung, ob die PIN-Eingabe korrekt war. Anhand N_{S2} kann der Authentication-Server prüfen, ob die Antwort wirklich gerade gesendet wurde. Wenn der Authentication-Server von der Authentizität des Benutzers überzeugt ist, dann generiert er den Schlüssel K_{MP} mir Hilfe einer zufälligen Zahl. Das mobile Gerät erhält diesen Schlüssel durch eine letzte Nachricht, bei der K_{MP} zusammen mit $N_M + 1$, mit dem Schlüssel K_{MS} verschlüsselt übertragen wird. N_M wird um 1 erhöht, damit diese Nachricht von der Nachricht (14) unterschieden werden kann. Ansonsten könnte ein Angreifer nach der Kompromittierung des Schlüssels K_{CM}, die Nachricht (14) anstatt der Nachricht (20) einspielen und so den Schlüssel K_{MP} auf der Seite des mobilen Geräts mit K_{CM} gleichsetzen.
(20) $S \longrightarrow M : \{K_{MP}, N_M + 1\}_{K_{MS}}$

Das mobile Gerät entschlüsselt die Nachricht und prüft $N_M + 1$. Wenn $N_M + 1$ korrekt ist, dann hat das mobile Gerät den aktuellen Schlüssel K_{MP} und die EAP-Authentifizierung ist damit erfolgreich abgeschlossen.

K_{MP} ist das Ergebnis der Authentifizierung und kann anschließend als PMK von der RSN-Sicherheitsarchitektur, wie in Kapitel 2.4.1 beschrieben, verwendet werden.

4.2.2 Optimierung des Nachrichtenablaufs

Die aus dem vorherigen Kapitel ermittelten Nachrichten werden jetzt so optimiert, dass der Nachrichtenablauf minimiert ist, ohne die Funktionalität der EAP-CHIP-Authentifizierung zu verändern. Dadurch werden Angriffsmöglichkeiten zwischen den einzelnen Nachrichten eingeschränkt.

Um die Optimierung zu erreichen, werden zunächst die Nachrichten tabellarisch aufgelistet (siehe Tabelle 4.1). Für jede Nachricht muss sowohl der Sender als auch der Empfänger angegeben werden. Weiterhin ist es notwendig, für jede Nachricht die notwendigen Vorgänger- und Nachfolgernachrichten zu ermitteln. Eine Nachricht, die als Vorgänger einer anderen Nachricht ermittelt wurde, ist automatisch ein Nachfolger dieser Nachricht.

Als erste Nachricht hat die Nachricht (01) keinen Vorgänger. Die Nachricht (02) benötigt ebenfalls keine vorherige Nachricht. Für das Senden der Nachricht (03) benötigt S den Wert aus Nachricht (02) und den Schlüssel K_{MS}, der aus der Nachricht (01) abgeleitet werden kann. Die Nachricht (04) soll S nur senden, wenn S bereits die Nachricht (01) erhalten hat. Die Nachricht (05) ist die Antwort auf die Nachricht (04).

C soll die Nachricht (06) erst senden, wenn S seine Authentizität gegenüber M bewiesen hat. Dies erfolgt durch die Nachricht (03). Die Nachricht (07) leitet die Nachricht (06) weiter an S. Die Nachricht (08) kann erst nach der Nachricht (06) verarbeitet werden, da dies ein anderer Funktionsaufruf der Chipkarte ist (siehe Kapitel 4.2.1). Die Nachricht (09) leitet die Nachricht (08) weiter an S. Damit S die Nachricht (10) senden kann, müssen die Informationen zunächst durch die Nachrichten (07) und (09) an S übermittelt worden sein, und M muss seine Authentizität gegenüber S bewiesen haben, was S nach der Nachricht (05) prüfen kann. Durch die Nachricht (11) wird die Nachricht (10) an C weitergeleitet. Mit der Nachricht (12) antwortet C auf die Nachricht (11). Die Weiterleitung der Nachricht (12) an S erfolgt durch die Nachricht (13). Nach dem Empfang der Nachricht (13) haben sich C und S gegenseitig authentifiziert und S sendet Nachricht (14) an M.

Erst wenn Nachricht (14) empfangen wurde, ist die Nachricht (15) zu senden. Die Nachricht (16) kann erst nach Nachricht (07) gesendet werden, weil S dann K_{CS} kennt. Die Nachricht (17) ist die Weiterleitung der Nachricht (16) an C und kann erst nach der erfolgreichen PIN-Prüfung nach Nachricht (15) gesendet werden. C antwortet mit der Nachricht (18) auf die Nachricht (17). Durch die Nachricht (19) erhält S die Nachricht (18). Die Nachricht (20) ist die letzte Nachricht und wird von S erst gesendet, wenn dieser zuvor die Nachricht (19) empfangen hat.

Nun werden die Nachrichten mit demselben Empfänger und Sender zusammengefasst, wenn mindestens ein Vorgänger bzw. Nachfolger übereinstimmt. Die Vorgänger und Nachfolger der neuen Nachricht entsprechen den vereinigten Mengen der zugrunde liegenden Nachrichten.

Tabelle 4.1: Abhängigkeiten der Nachrichten

Nr.	Nachricht	Sender	Empfänger	Vorgänger	Nachfolger
(01)	ID_M	M	S	-	(03),(04)
(02)	N_M	M	S	-	(03)
(03)	$\{N_M\}_{K_{MS}}$	S	M	(01),(02)	(06)
(04)	N_{S1}	S	M	(01)	(05)
(05)	$\{N_{S1}\}_{K_{MS}}$	M	S	(04)	(10)
(06)	ID_C	C	M	(03)	(07),(08)
(07)	$\{ID_C\}_{K_{MS}}$	M	S	(06)	(10),(16)
(08)	N_C	C	M	(06)	(09)
(09)	$\{N_C\}_{K_{MS}}$	M	S	(08)	(10)
(10)	$\{N_{S2}, N_C, ID_C\}_{K_{CS}}$	S	M	(07),(09),(05)	(11)
(11)	$\{N_{S2}, N_C, ID_C\}_{K_{CS}}$	M	C	(10)	(12)
(12)	$\{N_C, N_{S2}\}_{K_{CS}}$	C	M	(11)	(13)
(13)	$\{N_C, N_{S2}\}_{K_{CS}}$	M	S	(12)	(14)
(14)	$\{K_{CM}, N_M\}_{K_{MS}}$	S	M	(13)	(15)
(15)	$\{PIN\}_{K_{CM}}$	M	C	(14)	(17)
(16)	$\{N_{S2}\}_{K_{CS}}$	S	M	(07)	(17)
(17)	$\{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}$	M	C	(15),(16)	(18)
(18)	$\{\{A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}$	C	M	(17)	(19)
(19)	$\{A, N_{S2}\}_{K_{CS}}$	M	S	(18)	(20)
(20)	$\{K_{MP}, N_M + 1\}_{K_{MS}}$	S	M	(19)	-

Die Nachrichten (01) und(02) besitzen keine Vorgängernachrichten. Durch das Zusammenfassen dieser Nachrichten ergibt sich die folgende Nachricht:
(01) +(02) $M \longrightarrow S : ID_M, N_M$

Als nächstes haben die Nachrichten (03) und (04) denselben Vorgänger (01). Da der Dateninhalt von Nachricht (04) unverschlüsselt ist, kann dieser mit verschlüsselt werden, weil M der Schlüssel K_{MS} bekannt ist. Das führt zu der folgenden Nachricht:
(03) +(04) $S \longrightarrow M : \{N_M, N_{S1}\}_{K_{MS}}$

Eine weitere Möglichkeit des Zusammenfügens von Nachrichten ergibt sich durch die Nachrichten (05), (07) und (09), die denselben Nachfolger (10) besitzen. Alle drei Nachrichten sind mit K_{MS} verschlüsselt, der dem Sender M bekannt ist. Das ermöglicht, alle Inhalte gemeinsam zu verschlüsseln. Das ergibt die folgende Nachricht:
(05) +(07) +(09) $M \longrightarrow S : \{N_{S1}, ID_C, N_C\}_{K_{MS}}$

Als Letztes können die Nachrichten (10) und (16) zusammengefügt werden, weil beide die Vorgängernachricht (07) haben. Beide Nachrichten sind mit K_{CS} verschlüsselt, aber M ist

dieser Schlüssel nicht bekannt. Somit müssen sie weiter einzeln verschlüsselt sein.
(10) +(16) $M \longrightarrow S : \{N_{S2}, N_C, ID_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}}$

Alle verbleibenden Nachrichten bleiben erhalten und ordnen sich entsprechend ihrer Vorgänger und Nachfolger in die Nachrichtenfolge ein.

Die Tabelle 4.2 zeigt die neue Nachrichtenfolge. Die neue Nachricht (01) entspricht der Kombination aus den alten Nachrichten (01) und (02) und übernimmt die Position am Anfang der Nachrichtenfolge. Dieser folgt die neue Nachricht (02), die aus den alten Nachrichten (03) und (04) zusammengesetzt ist und diese auch ersetzt. Die Kombination der alten Nachrichten (05), (07) und (09) ergibt die neue Nachricht (05), und übernimmt die Position der alten Nachricht (09), weil beide den Nachfolger (10), aber unterschiedliche Vorgänger haben. Anschließend folgt die neue Nachricht (06), was die Kombination von Nachricht (10) und (16) ist, weil beide Ursprungsnachrichten die alte Nachricht (07) als Vorgänger haben.

Tabelle 4.2: Optimierte Nachrichtenfolge

Neu-Nr.	Nachricht	Sender	Empfänger	Alt-Nr.
(01)	ID_M, N_M	M	S	(01),(02)
(02)	$\{N_M, N_{S1}\}_{K_{MS}}$	S	M	(03),(04)
(03)	ID_C	C	M	(06)
(04)	N_C	C	M	(08)
(05)	$\{N_{S1}, ID_C, N_C\}_{K_{MS}}$	M	S	(05),(07),(09)
(06)	$\{N_{S2}, N_C, ID_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}}$	S	M	(10),(16)
(07)	$\{N_{S2}, N_C, ID_C\}_{K_{CS}}$	M	C	(11)
(08)	$\{N_C, N_{S2}\}_{K_{CS}}$	C	M	(12)
(09)	$\{N_C, N_{S2}\}_{K_{CS}}$	M	S	(13)
(10)	$\{K_{CM}, N_M\}_{K_{MS}}$	S	M	(14)
(11)	$\{PIN\}_{K_{CM}}$	M	C	(15)
(12)	$\{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}$	M	C	(17)
(13)	$\{\{A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}$	C	M	(18)
(14)	$\{A, N_{S2}\}_{K_{CS}}$	M	S	(19)
(15)	$\{K_{MP}, N_M + 1\}_{K_{MS}}$	S	M	(20)

Zusammenfassend zeigt die Abbildung 4.2 den Ablauf einer erfolgreichen EAP-CHIP-Authentifizierung. Die Aktionen der jeweiligen Komponenten C, M und S, sind in der entsprechenden Spalte in ihrer zeitlichen Abfolge zusammengefasst. Nachrichten sind als Richtungspfeile zwischen den Komponenten dargestellt. Eine Nachricht muss direkt nach einer Aktion beginnen und endet immer vor einer Aktion einer anderen Komponente. Die Aktion, bei der eine Nachricht endet, startet unmittelbar nach Empfang der Nachricht. Aktionen mit einer „ISO/IEC“-Markierung sind von den Standards ISO/IEC 7816-4/8 der Chipkarte abhängig. Zur Vervollständigung wurden bei den Nachrichten $C \longrightarrow M : ID_C$, $C \longrightarrow M : N_C$

und $C \longrightarrow M : \{PIN\}_{K_{CM}}$ jeweils eine Nachricht in anderer Richtung eingefügt. Das soll verdeutlichen, dass es sich um Aufrufe von Chipkartenfunktionen handelt.

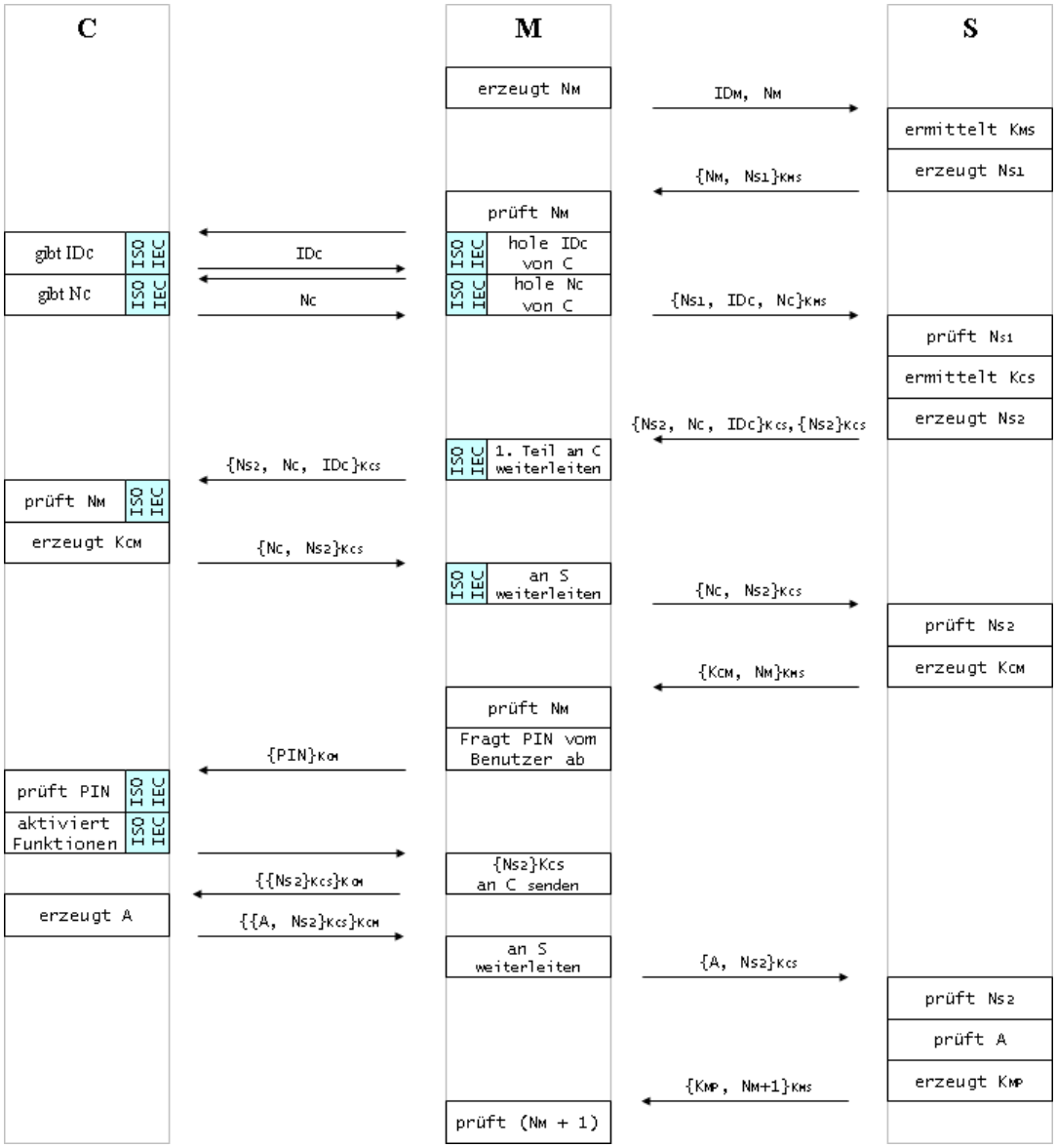

Abbildung 4.2: Ablauf einer erfolgreichen EAP-CHIP-Authentifizierung

5 Verifizierung des Entwurfs

Im vorherigen Kapitel wurde eine Sicherheitsarchitektur mit dem neuen Authentifizierungsverfahren EAP-CHIP entworfen. Nun ist es notwendig zu prüfen, ob das EAP-CHIP-Protokoll korrekt arbeitet und die Sicherheitsanforderungen erfüllt werden. Alle anderen Verfahren, die Bestandteil der Sicherheitsarchitektur sind, werden nicht weiter untersucht. Dies würde sonst den Rahmen der Arbeit sprengen.

Zur Untersuchung des Protokolls eignet sich die BAN-Logik, die einen Formalismus zur Analyse von Authentifizierungsprotokollen bereitstellt (Eckert 2003).

5.1 BAN-Logik

Ein Authentifizierungsprotokoll ist eine Abfolge von Nachrichten zwischen Principals, die die jeweiligen Kommunikationspartner repräsentieren. Jede Nachricht wird dabei durch Symbole dargestellt, die den Sender, den Empfänger und den Nachrichteninhalt repräsentieren. Die konventionelle Notation ist für eine logische Analyse nicht geeignet, weil dafür die exakte Bedeutung der jeweiligen Nachrichten benötigt wird. Diese Bedeutungen sind aber nicht immer Bestandteil des Nachrichteninhalts. Für eine bessere Darstellung des Protokolls wird jede Nachricht unter Verwendung der Protokollbeschreibung in eine logische Formel umgewandelt. Diese logische Formel ist die idealisierte Version der originalen Nachricht. Zusätzlich werden Behauptungen hinzugefügt, die zum jeweiligen Protokollschritt beschreiben, woran die beteiligten Principals glauben (engl. *belief*) (Burrows u. a. 1990).

5.1.1 Notation

Die BAN-Logik unterscheidet drei Arten von Objekten: Principals, kryptographische Schlüssel und Formeln. Die Objekte werden durch Symbole repräsentiert. Für jedes Objekt sind Symbole festgelegt, die z. B. als Metasymbole verwendet werden können. Für Principals sind es die Symbole P, Q und R, für Schlüssel ist es das Symbol K und für Formeln die Symbole X und Y.

Um Objekte zu symbolisieren, werden für Principals generell die Symbole A, B und S verwendet. Ein symmetrischer Schlüssel, der von 2 Principals, z. B. A und B, verwendet wird, wird normalerweise in der Form K_{AB} dargestellt. Spezifische Ausdrücke werden durch Symbole wie N_A, N_B usw. repräsentiert.

Zur Beschreibung der logischen Formeln werden Operatoren verwendet wie in Tabelle 5.1 aufgeführt. Wie diese Operatoren verwendet werden, wird in den folgenden Abschnitten ersichtlich.

Tabelle 5.1: BAN-Operatoren

Operator	Beschreibung
$P \mid\equiv X$	P glaubt an die Formel X.
$P \lhd X$	P hat die Formel X empfangen.
$P \mid\sim X$	P hat die Formel X irgendwann gesendet, und war vom Wahrheitsgehalt überzeugt.
$\#(X)$	Die Formel X ist frisch, d.h. sie wurde nicht vor diesem Protokolllauf gesendet.
$P \mid\Rightarrow X$	P kontrolliert X, d.h. P besitzt die Autorität, den Wahrheitsgehalt von X zu bestätigen.
$P \overset{K}{\leftrightarrow} Q$	K ist der symmetrische Schlüssel für die Kommunikation zwischen P und Q. Die Formel $Q \overset{K}{\leftrightarrow} P$ symbolisiert den selben Schlüssel.
$\{X\}_K$	Die Formel X wurde mit dem Schlüssel K verschlüsselt.
(X, Y)	Die Formel (X, Y) besteht aus den Teilformeln X und Y.

5.1.2 Ableitungsregeln

Um neue Erkenntnisse in der Analyse zu erhalten, wurden Regeln festgelegt. Diese Regeln ermöglichen es, von bekannten logischen Formeln neue Formeln abzuleiten. Es werden hier nur die Regeln beschrieben, die in dieser Analyse benötigt werden. Weitere Regeln sind im Originaldokument (Burrows u. a. 1990) formuliert.

- *Message-Meaning*-Regel:

  $$\frac{P \mid\equiv P \overset{K}{\leftrightarrow} Q, \quad P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$$

  Wenn P glaubt, dass der Schlüssel K der symmetrische Schlüssel mit Q ist und P die mit K verschlüsselte Nachricht X empfangen hat, dann glaubt P, dass Q die Nachricht X irgendwann gesendet hat.

- *Nonce-Verification*-Regel:

  $$\frac{P \mid\equiv \#(X),\ \ P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

  Wenn P glaubt, dass X frisch ist und X von Q irgendwann gesendet wurde, dann glaubt P, dass Q an die Formel X glaubt. Diese Regel findet ihre Anwendung bei Challenge-Response-Verfahren. Ein *Challenge* ist ein frischer Wert, der bei einer Antwortnachricht verschlüsselt sein muss.

- *Jurisdiction*-Regel:

  $$\frac{P \mid\equiv Q \mid\Rightarrow X,\ \ P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

  Wenn P glaubt, dass X von Q kontrolliert wird und Q an X glaubt, dann glaubt P an den Wahrheitsgehalt von X. Diese Regel wird z. B. zur Prüfung eines empfangenen Schlüssels benötigt.

- $\mid\equiv$-Regel:

  $$\frac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv X}, \frac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv Y}$$

  Wenn P glaubt, dass Q an die Formel (X,Y) glaubt, dann glaubt P auch, dass Q an die jeweiligen Teilformeln X und Y glaubt.

- $\mid\sim$-Regel:

  $$\frac{P \mid\equiv Q \mid\sim (X,Y)}{P \mid\equiv Q \mid\sim X}, \frac{P \mid\equiv Q \mid\sim (X,Y)}{P \mid\equiv Q \mid\sim Y}$$

  Wenn P glaubt, dass die Formel (X,Y) von Q gesendet wurde, dann glaubt P auch, dass X bzw. Y von Q gesendet wurde.

- $\lhd$-*Parting*-Regel:

  $$\frac{P \lhd (X,Y)}{P \lhd X}, \frac{P \lhd (X,Y)}{P \lhd Y}$$

  Wenn P die Formel (X,Y) empfangen hat, dann hat P auch X bzw. Y empfangen.

- Empfangsregel:

  $$\frac{P \mid\equiv P \overset{K}{\leftrightarrow} Q,\ \ P \lhd \{X\}_K}{P \lhd X}$$

  Wenn P an den symmetrischen Schlüssel K zwischen P und Q glaubt und P die mit K verschlüsselte Formel X empfangen hat, dann hat P auch X empfangen. Diese Regel entspricht dem Entschlüsseln einer Nachricht X.

- Frische-Regel:

$$\frac{P \mid\equiv \#(X),\ \ P \lhd (X,Y)}{P \mid\equiv \#(X,Y)}\ ,\ \frac{P \mid\equiv \#(Y),\ \ P \lhd (X,Y)}{P \mid\equiv \#(X,Y)}$$

  Wenn P an die Frische der Formel X bzw. Y glaubt und P die Formel (X,Y) empfangen hat, dann glaubt P auch an die zusammengefasste Formel (X,Y).

5.1.3 Idealisiertes Protokoll

Für die BAN-Logik muss jeder Protokollschritt in eine idealisierte Form gebracht werden. Eine Nachricht im idealisierten Protokoll entspricht einer Formel. Klartextnachrichten und identifizierende Ausdrücke entfallen, weil diese nicht relevant für die Sicherheit des Protokolls sind. Schlüssel werden in einen Schlüsseloperator entsprechend ihrer Funktion transformiert. Sonstige Ausdrücke werden der Notation entsprechend angepasst.

5.1.4 Ablauf der Analyse

Für Analyse mit der BAN-Logik sind folgende Analyseschritte notwendig.

- Das idealisierte Protokoll ist vom Originalprotokoll abzuleiten.
- Die Annahmen über die Anfangszustände zum Protokollstart sind festzulegen.
- Die Zielbehauptungen des Protokolls sind festzulegen.
- Zu jedem Protokollschritt (jeder idealisierten Nachricht) sind neue Behauptungen zu ermitteln. Die Behauptung $P \lhd X$ kann immer direkt von der Nachricht X, die P empfangen hat, festgelegt werden.

Wenn eine Behauptung aufgestellt oder abgeleitet wird, dann ist diese Behauptung für alle folgenden Protokollschritte gültig. Die Regeln werden so angewendet, dass die Ziele zu einem möglichst frühen Zeitpunkt erreicht werden. Ableitungen, die keine neuen Erkenntnisse liefern, sollten vermieden werden, damit man die Übersicht behalten kann.

5.2 Idealisierung des Protokolls

Als Grundlage für die Idealisierung wird der optimierte Nachrichtenablauf von EAP-CHIP verwendet, der in Tabelle 4.2 aufgelistet ist. Die Symbole für die Principals entsprechen den Symbolen der Protokollteilnehmer, was die Idealisierung vereinfacht. Somit repräsentieren C, M und S die Principals.

Die Nachrichten (01), (03) und (04) entfallen, weil diese nur Klartext-Informationen enthalten. Die Nachricht (02) kann unmittelbar übernommen werden, weil diese der BAN-Notation entspricht. Aus den Nachrichten (05), (06) und (07) wird ID_C entfernt, da dies nur ein identifizierender Ausdruck ist. Die Nachrichten (08) und (09) werden wieder unmittelbar übernommen. In der Nachricht (10) wird der Schlüssel K_{CM} transformiert in $C \overset{K_{CM}}{\longleftrightarrow} M$. In der Nachricht (11) wird der Wert PIN zu N_P. Die Nachricht (12) wird wieder unmittelbar übernommen. In den Nachrichten (13) und (14) wird die Antwort A zu N_A. In der Nachricht (15) wird K_{MP} transformiert in $M \overset{K_{MP}}{\longleftrightarrow} P$ und aus $N_M + 1$ wird N_M, weil $N_M + 1$ im unmittelbaren Zusammenhang zum Wert N_M steht. Die Tabelle 5.2 zeigt das gesamte idealisierte Protokoll.

Tabelle 5.2: Idealisiertes EAP-CHIP-Protokoll

Nr.	Nachricht
(01)	-
(02)	$S \longrightarrow M : \{N_M, N_{S1}\}_{K_{MS}}$
(03)	-
(04)	-
(05)	$M \longrightarrow S : \{N_{S1}, N_C\}_{K_{MS}}$
(06)	$S \longrightarrow M : \{N_{S2}, N_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}}$
(07)	$M \longrightarrow C : \{N_{S2}, N_C\}_{K_{CS}}$
(08)	$C \longrightarrow M : \{N_C, N_{S2}\}_{K_{CS}}$
(09)	$M \longrightarrow S : \{N_C, N_{S2}\}_{K_{CS}}$
(10)	$S \longrightarrow M : \{C \overset{K_{CM}}{\longleftrightarrow} M, N_M\}_{K_{MS}}$
(11)	$M \longrightarrow C : \{N_P\}_{K_{CM}}$
(12)	$M \longrightarrow C : \{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}$
(13)	$C \longrightarrow M : \{\{N_A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}$
(14)	$M \longrightarrow S : \{N_A, N_{S2}\}_{K_{CS}}$
(15)	$S \longrightarrow M : \{M \overset{K_{MP}}{\longleftrightarrow} P, N_M\}_{K_{MS}}$

5.3 Protokollannahmen

Die folgenden Formeln entsprechen den Annahmen zu Beginn einer EAP-CHIP-Authentifizierung.

- K_{MS} ist der symmetrische Schlüssel zwischen M und S. Somit müssen zu Beginn beide Principals an diesen Schlüssel glauben.

  Annahme 1: $M \mid\equiv M \overset{K_{MS}}{\leftrightarrow} S$

  Annahme 2: $S \mid\equiv M \overset{K_{MS}}{\leftrightarrow} S$

- K_{CS} ist der symmetrische Schlüssel zwischen C und S. Folglich müssen auch diese beiden Principals an diesen Schlüssel glauben.

  Annahme 3: $C \mid\equiv C \overset{K_{CS}}{\leftrightarrow} S$

  Annahme 4: $S \mid\equiv C \overset{K_{CS}}{\leftrightarrow} S$

- K_{CM} ist der symmetrische Schlüssel zwischen C und M, der von C und S gleichzeitig erzeugt wird. C glaubt somit selbst an diesen Schlüssel, und M muss daran glauben, dass S einen gültigen Schlüssel erzeugen kann.

  Annahme 5: $C \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M$

  Annahme 6: $M \mid\equiv S \mid\Rightarrow C \overset{K_{CM}}{\leftrightarrow} M$

- Alle Principals glauben daran, dass ihre Zufallszahlen für jeden Protokolllauf neu generiert werden und somit frisch sind.

  Annahme 7: $C \mid\equiv \#(N_C)$

  Annahme 8: $M \mid\equiv \#(N_M)$

  Annahme 9: $S \mid\equiv \#(N_{S1})$

  Annahme 10: $S \mid\equiv \#(N_{S2})$

- Der symmetrische Schlüssel K_{MP} ist das Ergebnis einer erfolgreichen Authentifizierung. M erhält diesen Schlüssel von S und deshalb muss M daran glauben, dass S einen gültigen Schlüssel erzeugen kann.

  Annahme 11: $M \mid\equiv S \mid\Rightarrow M \overset{K_{MP}}{\leftrightarrow} P$

5.4 Protokollziele

Nachstehend aufgeführte Protokollziele sollen im Verlauf der Authentifizierung erreicht werden. Für die Bestimmung dieser Ziele wurden die ermittelten Sicherheitsanforderungen zu Grunde gelegt. Wenn alle Ziele erreicht werden, arbeitet das Protokoll korrekt. Zusätzlich wird angegeben, nach welchem Protokollschritt das jeweilige Ziel erreicht werden muss.

- Damit M überzeugt ist, dass S authentisch ist, muss M wissen, ob S an den Zufallswert N_M von M glaubt. Dieses Ziel entspricht nach (Burrows u. a. 1990) dem Ziel für Challenge-Response-Verfahren. Dieses Ziel muss nach Protokollschritt (02) erreicht worden sein.

  Ziel 1: $M \mid\equiv S \mid\equiv N_M$

- S muss davon überzeugt sein, dass M authentisch ist. Somit muss S wissen, dass M an den Zufallswert N_{S1} glaubt. Nach Protokollschritt (05) soll dieses Ziel erreicht sein.

  Ziel 2: $S \mid\equiv M \mid\equiv N_{S1}$

- C muss von der Authentizität von S überzeugt sein. Deshalb muss C wissen, ob S an den Zufallswert N_C glaubt. Mit Protokollschritt (07) wird N_C verschlüsselt an C gesendet.

  Ziel 3: $C \mid\equiv S \mid\equiv N_C$

- S muss auch von der Authentizität von C überzeugt sein, weshalb S wissen muss, ob C an den Zufallswert N_{S2} glaubt. Protokollschritt (09) beinhaltet den verschlüsselten Wert N_{S2}.

  Ziel 4: $S \mid\equiv C \mid\equiv N_{S2}$

- Da M den Schlüssel K_{CM} von S erhält, muss M glauben das K_{CM} wirklich der aktuelle Schlüssel ist. M erfährt den Schlüssel K_{CM} durch Protokollschritt (10).

  Ziel 5: $M \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M$

- C muss glauben, dass M eine PIN N_P gesendet hat, an die auch M glaubt. C erhält N_P mit dem Protokollschritt (11).

  Ziel 6: $C \mid\equiv M \mid\sim N_P$

- Da C mit der Antwort N_A bestätigt, dass die PIN korrekt ist, muss S davon überzeugt sein, dass auch C an N_A glaubt. Dies wird durch den Protokollschritt (14) erreicht.

  Ziel 7: $S \mid\equiv C \mid\equiv N_A$

- M soll am Ende des Protokolllaufs den Sitzungsschlüssel K_{MP} erhalten. Dies erfolgt durch den letzten Protokollschritt (15).

  Ziel 8: $M \mid\equiv M \overset{K_{MP}}{\leftrightarrow} P$

5.5 Protokollanalyse

Ausgehend von den Protokollannahmen werden nun für alle Protokollschritte Behauptungen so abgeleitet, dass die Protokollziele erreicht werden. Die Behauptungen werden durch zwei Zahlen gekennzeichnet. Die erste Zahl entspricht dem Protokollschritt, in dem die Behauptung aufgestellt bzw. abgeleitet wurde. Die zweite Zahl ist eine fortlaufende Nummerierung innerhalb des Protokollschritts.

Protokollschritt (02): $S \longrightarrow M : \{N_M, N_{S1}\}_{K_{MS}}$

Behauptung 2.1 : $M \lhd \{N_M, N_{S1}\}_{K_{MS}}$

Aus der **Behauptung 2.1** und der **Annahme 1** ergibt sich mittels der *Message-Meaning*-Regel die

Behauptung 2.2 : $$\frac{M \mid\equiv M \overset{K_{MS}}{\leftrightarrow} S,\quad M \lhd \{N_M, N_{S1}\}_{K_{MS}}}{M \mid\equiv S \mid\sim (N_M, N_{S1})}$$

Durch die Anwendung der $\mid\sim$-Regel folgt aus der **Behauptung 2.2** die folgende

Behauptung 2.3 : $$\frac{M \mid\equiv S \mid\sim (N_M, N_{S1})}{M \mid\equiv S \mid\sim N_M}$$

Wird nun die *Nonce-Verification*-Regel mit der **Annahme 8** und **Behauptung 2.3** angewendet, dann folgt daraus die

Behauptung 2.4 : $$\frac{M \mid\equiv \#(N_M),\quad M \mid\equiv S \mid\sim N_M}{M \mid\equiv S \mid\equiv N_M}$$

Die **Behauptung 2.5** entspricht dem **Ziel 1** $M \mid\equiv S \mid\equiv N_M$, was somit auch erreicht wurde.

Protokollschritt (05): $M \longrightarrow S : \{N_{S1}, N_C\}_{K_{MS}}$

Behauptung 5.1 : $S \lhd \{N_{S1}, N_C\}_{K_{MS}}$

Durch die *Message-Meaning*-Regel folgt aus der **Annahme 2** und der **Behauptung 5.1** die

Behauptung 5.2 : $$\frac{S \mid\equiv S \overset{K_{MS}}{\leftrightarrow} M,\quad S \lhd \{N_{S1}, N_C\}_{K_{MS}}}{S \mid\equiv M \mid\sim (N_{S1}, N_C)}$$

Wird nun die $|\sim$-Regel mit der **Behauptung 5.2** angewendet, dann folgen daraus die

Behauptung 5.3 : $\dfrac{S \ |\equiv \ M \ |\sim \ (N_{S1}, N_C)}{S \ |\equiv \ M \ |\sim \ N_{S1}}$

und die

Behauptung 5.4 : $\dfrac{S \ |\equiv \ M \ |\sim \ (N_{S1}, N_C)}{S \ |\equiv \ M \ |\sim \ N_C}$

Durch die Anwendung der *Nonce-Verification*-Regel ergibt sich aus der **Behauptung 5.3** und der **Annahme 9** die

Behauptung 5.5 : $\dfrac{S \ |\equiv \ \#(N_{S1}), \ \ S \ |\equiv \ M \ |\sim \ N_{S1}}{S \ |\equiv \ M \ |\equiv \ N_{S1}}$

Die **Behauptung 5.5** stimmt mit dem **Ziel 2** $S \ |\equiv \ M \ |\equiv \ N_{S1}$ überein.

Protokollschritt (06): $S \longrightarrow M : \{N_{S2}, N_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}}$

Behauptung 6.1 : $M \ \lhd \ \{N_{S2}, N_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}}$

Wird die $\lhd$-*Parting*-Regel auf die **Behauptung 6.1** angewendet, dann ergeben sich daraus die

Behauptung 6.2 : $\dfrac{M \ \lhd \ (\{N_{S2}, N_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}})}{M \ \lhd \ \{N_{S2}, N_C\}_{K_{CS}}}$

und die

Behauptung 6.3 : $\dfrac{M \ \lhd \ (\{N_{S2}, N_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}})}{M \ \lhd \ \{N_{S2}\}_{K_{CS}}}$

Da M nicht den Schlüssel K_{CS} kennt, können keine weiteren Behauptungen abgeleitet werden. Das entspricht der Logik des Protokolls, da diese Nachricht nicht für M bestimmt ist.

Protokollschritt (07): $M \longrightarrow C : \{N_{S2}, N_C\}_{K_{CS}}$

Behauptung 7.1 : $C \ \lhd \ \{N_{S2}, N_C\}_{K_{CS}}$

Durch die *Message-Meaning*-Regel ergibt sich aus der **Annahme 3** und der **Behauptung 7.1** die

Behauptung 7.2 : $\dfrac{C \ |\equiv \ C \overset{K_{CS}}{\leftrightarrow} S, \ \ C \ \lhd \ \{N_{S2}, N_C\}_{K_{CS}}}{C \ |\equiv \ S \ |\sim \ (N_{S2}, N_C)}$

Durch die $|\sim$-Regel folgt aus der **Behauptung 7.2** die

Behauptung 7.3 : $\dfrac{C \ |\equiv \ S \ |\sim \ (N_{S1}, N_C)}{C \ |\equiv \ S \ |\sim \ N_C}$

Anschließend ergibt sich aus der **Annahme 7** und der **Behauptung 7.3** durch Anwendung der *Nonce-Verification*-Regel, die

Behauptung 7.4 : $$\frac{C \mid\equiv \#(N_C),\ \ C \mid\equiv S \mid\sim N_C}{C \mid\equiv S \mid\equiv N_C}$$

Damit wurde das **Ziel 3** $C \mid\equiv S \mid\equiv N_C$ erreicht, was der **Behauptung 7.4** entspricht.

Protokollschritt (08): $C \longrightarrow M : \{N_C, N_{S2}\}_{K_{CS}}$

Behauptung 8.1 : $M \lhd \{N_C, N_{S2}\}_{K_{CS}}$

M ist der Schlüssel K_{CS} nicht bekannt. Somit können keine weiteren Behauptungen abgeleitet werden.

Protokollschritt (09): $M \longrightarrow S : \{N_C, N_{S2}\}_{K_{CS}}$

Behauptung 9.1 : $S \lhd \{N_C, N_{S2}\}_{K_{CS}}$

Durch die *Message-Meaning*-Regel folgt aus der **Annahme 4** und der **Behauptung 9.1** die

Behauptung 9.2 : $$\frac{S \mid\equiv S \overset{K_{CS}}{\longleftrightarrow} C,\ \ S \lhd \{N_C, N_{S2}\}_{K_{CS}}}{S \mid\equiv C \mid\sim (N_C, N_{S2})}$$

Nun wird die $\mid\sim$-Regel auf die **Behauptung 9.2** angewendet und es folgt die

Behauptung 9.3 : $$\frac{S \mid\equiv C \mid\sim (N_C, N_{S2})}{S \mid\equiv C \mid\sim N_{S2}}$$

Durch die *Nonce-Verification*-Regel, angewendet auf die **Annahme 10** und die **Behauptung 9.3** , ergibt sich die

Behauptung 9.4 : $$\frac{S \mid\equiv \#(N_{S2}),\ \ S \mid\equiv C \mid\sim N_{S2}}{S \mid\equiv C \mid\equiv N_{S2}}$$

Damit wird das **Ziel 4** erreicht, weil die **Behauptung 9.4** diesem Ziel entspricht.

Protokollschritt (10): $S \longrightarrow M : \{C \overset{K_{CM}}{\longleftrightarrow} M, N_M\}_{K_{MS}}$

Behauptung 10.1 : $M \lhd \{C \overset{K_{CM}}{\longleftrightarrow} M, N_M\}_{K_{MS}}$

Unter Anwendung der *Message-Meaning*-Regel ergibt sich aus der **Annahme 1** und der **Behauptung 10.1** die

Behauptung 10.2 : $$\frac{M \mid\equiv M \overset{K_{MS}}{\longleftrightarrow} S,\ \ M \lhd \{C \overset{K_{CM}}{\longleftrightarrow} M, N_M\}_{K_{MS}}}{M \mid\equiv S \mid\sim (C \overset{K_{CM}}{\longleftrightarrow} M, N_M)}$$

Zusätzlich erhält man durch die Empfangsregel die

Behauptung 10.3 : $\dfrac{M \mid\equiv M \overset{K_{MS}}{\leftrightarrow} S,\quad M \lhd \{C \overset{K_{CM}}{\leftrightarrow} M, N_M\}_{K_{MS}}}{M \lhd (C \overset{K_{CM}}{\leftrightarrow} M, N_M)}$

Wird die Frische-Regel mit der **Annahme 8** und der **Behauptung 10.3** angewendet, dann erhält man die

Behauptung 10.4 : $\dfrac{M \mid\equiv \#(N_M),\quad M \lhd (C \overset{K_{CM}}{\leftrightarrow} M, N_M)}{M \mid\equiv \#(C \overset{K_{CM}}{\leftrightarrow} M, N_M)}$

Nun wird die *Nonce-Verification*-Regel angewendet. Dafür werden die **Behauptung 10.4** und die **Behauptung 10.2** zu Grunde gelegt. Das ergibt die

Behauptung 10.5 : $\dfrac{M \mid\equiv \#(C \overset{K_{CM}}{\leftrightarrow} M, N_M),\quad M \mid\equiv S \mid\sim (C \overset{K_{CM}}{\leftrightarrow} M, N_M)}{M \mid\equiv S \mid\equiv (C \overset{K_{CM}}{\leftrightarrow} M, N_M)}$

Auf diese neue Behauptung wird die $\mid\equiv$-Regel angewendet. Damit erhält man die

Behauptung 10.6 : $\dfrac{M \mid\equiv S \mid\equiv (C \overset{K_{CM}}{\leftrightarrow} M, N_M)}{M \mid\equiv S \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M}$

Wird nun die *Jurisdiction*-Regel auf die **Annahme 6** und die **Behauptung 10.6** angewendet, dann resultiert daraus die

Behauptung 10.7 : $\dfrac{M \mid\equiv S \mid\Rightarrow C \overset{K_{CM}}{\leftrightarrow} M,\quad M \mid\equiv S \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M}{M \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M}$

Damit ist das **Ziel 5** $M \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M$ erreicht.

Protokollschritt (11): $M \longrightarrow C : \{N_P\}_{K_{CM}}$

Behauptung 11.1 : $C \lhd \{N_P\}_{K_{CM}}$

Wenn die *Message-Meaning*-Regel auf die **Annahme 5** und die **Behauptung 11.1** angewendet wird, dann ergibt das die

Behauptung 11.2 : $\dfrac{C \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M,\quad C \lhd \{N_P\}_{K_{CM}}}{C \mid\equiv M \mid\sim N_P}$

Diese Behauptung entspricht dem **Ziel 6** $C \mid\equiv M \mid\sim N_P$, was damit erfüllt ist.

Protokollschritt (12): $M \longrightarrow C : \{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}$

Behauptung 12.1 : $C \lhd \{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}$

Durch das Anwenden der Empfangsregel auf die **Annahme 5** und die **Behauptung 12.1** ergibt sich die

Behauptung 12.2 : $\dfrac{C \mid\equiv C \overset{K_{CM}}{\leftrightarrow} M,\quad C \lhd \{\{N_{S2}\}_{K_{CS}}\}_{K_{CM}}}{C \lhd \{N_{S2}\}_{K_{CS}}}$

Durch das anschließende Anwenden der Empfangsregel auf die **Annahme 3** und die **Behauptung 12.2** ergibt sich die

Behauptung 12.3 : $$\frac{C \mid\equiv C \overset{K_{CS}}{\leftrightarrow} S,\ \ C \lhd \{N_{S2}\}_{K_{CS}}}{C \lhd N_{S2}}$$

Diese Behauptung dient zur Kontrolle, ob C auch N_{S2} erhält, welcher anschließend als Frischewert verwendet wird.

Protokollschritt (13): $C \longrightarrow M : \{\{N_A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}$

Behauptung 13.1 : $M \lhd \{\{N_A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}$

Durch die Empfangsregel folgt aus der **Behauptung 10.7** und der **Behauptung 13.1** die

Behauptung 13.2 : $$\frac{M \mid\equiv M \overset{K_{CM}}{\leftrightarrow} C,\ \ M \lhd \{\{N_A, N_{S2}\}_{K_{CS}}\}_{K_{CM}}}{M \lhd \{N_A, N_{S2}\}_{K_{CS}}}$$

Von dieser Behauptung sind keine weiteren Behauptungen ableitbar, weil M nicht den Schlüssel K_{CS} kennt. Dies ist korrekt, weil M diese mit K_{CS} verschlüsselte Nachricht weiter an S senden soll.

Protokollschritt (14): $M \longrightarrow S : \{N_A, N_{S2}\}_{K_{CS}}$

Behauptung 14.1 : $S \lhd \{N_A, N_{S2}\}_{K_{CS}}$

Wendet man die *Message-Meaning*-Regel auf die **Annahme 4** und die **Behauptung 14.1** an, dann ergibt das die

Behauptung 14.2 : $$\frac{S \mid\equiv S \overset{K_{CS}}{\leftrightarrow} C,\ \ S \lhd \{N_A, N_{S2}\}_{K_{CS}}}{S \mid\equiv C \mid\sim (N_A, N_{S2})}$$

Zusammen mit der **Annahme 10** erhält man nun durch die Anwendung der Frische-Regel die

Behauptung 14.3 : $$\frac{S \mid\equiv \#(N_{S2}),\ \ S \lhd (N_A, N_{S2})}{S \mid\equiv \#(N_A, N_{S2})}$$

Aus den letzten beiden Behauptungen und der *Nonce-Verification*-Regel ergibt sich die

Behauptung 14.4 : $$\frac{S \mid\equiv \#(N_A, N_{S2}),\ \ S \mid\equiv C \mid\sim (N_A, N_{S2})}{S \mid\equiv C \mid\equiv (N_A, N_{S2})}$$

Unter Anwendung der $\mid\equiv$-Regel erhält man jetzt die

Behauptung 14.5 : $$\frac{S \mid\equiv C \mid\equiv (N_A, N_{S2})}{S \mid\equiv C \mid\equiv N_A}$$

Die **Behauptung 14.5** entspricht dem **Ziel 7** $S \mid\equiv C \mid\equiv N_A$, das damit ebenfalls erreicht wurde.

Protokollschritt (15): $S \longrightarrow M : \{M \overset{K_{MP}}{\leftrightarrow} P, N_M\}_{K_{MS}}$

Behauptung 15.1 : $M \lhd \{M \overset{K_{MP}}{\leftrightarrow} P, N_M\}_{K_{MS}}$

Durch die *Message-Meaning*-Regel ergibt sich aus der **Annahme 1** und der **Behauptung 15.1** die

Behauptung 15.2 : $$\frac{M \mid\equiv M \overset{K_{MS}}{\leftrightarrow} S, \quad M \lhd \{M \overset{K_{MP}}{\leftrightarrow} P, N_M\}_{K_{MS}}}{M \mid\equiv S \mid\sim (M \overset{K_{MP}}{\leftrightarrow} P, N_M)}$$

Zusätzlich erhält man durch die Empfangsregel die

Behauptung 15.3 : $$\frac{M \mid\equiv M \overset{K_{MS}}{\leftrightarrow} S, \quad M \lhd \{M \overset{K_{MP}}{\leftrightarrow} P, N_M\}_{K_{MS}}}{M \lhd (M \overset{K_{MP}}{\leftrightarrow} P, N_M)}$$

Wird die Frische-Regel mit der **Annahme 8** und der **Behauptung 15.3** angewendet, dann ergibt das die

Behauptung 15.4 : $$\frac{M \mid\equiv \#(N_M), \quad M \lhd (M \overset{K_{MP}}{\leftrightarrow} P, N_M)}{M \mid\equiv \#(M \overset{K_{MP}}{\leftrightarrow} P, N_M)}$$

Nun wird die *Nonce-Verification*-Regel angewendet. Dafür werden die **Behauptung 15.4** und die **Behauptung 15.2** zu Grunde gelegt und das ergibt die

Behauptung 15.5 : $$\frac{M \mid\equiv \#(M \overset{K_{MP}}{\leftrightarrow} P, N_M), \quad M \mid\equiv S \mid\sim (M \overset{K_{MP}}{\leftrightarrow} P, N_M)}{M \mid\equiv S \mid\equiv (M \overset{K_{MP}}{\leftrightarrow} P, N_M)}$$

Auf diese neue Behauptung wird die $\mid\equiv$-Regel angewendet. Daraus folgt die

Behauptung 15.6 : $$\frac{M \mid\equiv S \mid\equiv (M \overset{K_{MP}}{\leftrightarrow} P, N_M)}{M \mid\equiv S \mid\equiv M \overset{K_{MP}}{\leftrightarrow} P}$$

Wird nun die *Jurisdiction*-Regel auf diese Behauptung und die **Annahme 11** angewendet, dann resultiert daraus die

Behauptung 15.7 : $$\frac{M \mid\equiv S \mid\Rightarrow M \overset{K_{MP}}{\leftrightarrow} P, \quad M \mid\equiv S \mid\equiv M \overset{K_{MP}}{\leftrightarrow} P}{M \mid\equiv M \overset{K_{MP}}{\leftrightarrow} P}$$

Die **Behauptung 15.7** stimmt mit dem **Ziel 8** $M \mid\equiv M \overset{K_{MP}}{\leftrightarrow} P$ überein, das damit auch erfüllt ist.

5.6 Ergebnis der Analyse

Durch die Analyse konnte festgestellt werden, dass alle acht Ziele durch die vorgesehenen Protokollschritte erreicht werden. Damit kann das Protokoll die Authentizitäten entsprechend den ermittelten Sicherheitsanforderungen prüfen.

Durch den Protokollschritt (13) tritt eine Redundanz auf, die sich durch eine unnötige Verschlüsselung bemerkbar macht. Die Ursache ist die Verwendung des *Secure Messaging*. Ob diese Redundanz für Angriffe ausgenutzt werden kann, liegt an den verwendeten Sicherheitsmechanismen. Beispielsweise kann ein beim *Secure Messaging* verwendeter Sendefolgezähler, wie in Kapitel 2.5.4 erwähnt, das Problem beheben.

In Eckert (2003) sind Eigenschaften der BAN-Logik beschrieben, die zeigen, dass eine solche Analyse nicht ausreicht. Denn eine Analyse mit der BAN-Logik ermöglicht nur Aussagen über wechselseitige Überzeugungen der Protokollteilnehmer, über Fragen der Delegierung und des Vertrauens in dritter Instanz (durch *Jurisdiction*-Regel) und über Möglichkeiten des Wiedereinspielens von Nachrichten (durch Frische-Regel) früherer Protokollläufe. Zusätzlich ermöglicht die BAN-Logik das Erkennen von Fehlern oder funktionalen Mängeln im frühen Stadium der Protokollentwicklung. Da sie eine Protokollanalyse nur auf einem sehr hohen Abstraktionsniveau ermöglicht, können keine Aussagen über die richtige Protokollimplementierung oder die verwendeten Verschlüsselungsalgorithmen gemacht werden. Weiterhin ist keine Aussage über die Verletzbarkeit des Protokolls durch passive Angriffe möglich. Außerdem fehlt der BAN-Logik eine formale semantische Fundierung der Logikkonstrukte. Daraus folgt, dass man keine Aussage über die Richtigkeit und Vollständigkeit der Analyse treffen kann. Für die weitere Entwicklung sollten deshalb weiterführende Analysemöglichkeiten eingesetzt werden. In Meadows (1995) wird ein Überblick über die wichtigsten formalen Methoden zur Verifikation von kryptographischen Protokollen gegeben (vgl. Eckert 2003, S. 444).

5.7 Erfahrungsbericht

Während der Entwicklung des Protokolls konnte die Analyse mit der BAN-Logik einige Probleme im Protokoll aufzeigen. Nachfolgend werden zwei dieser Probleme und ihre Lösungen beschrieben.

Fehlender Frische-Wert in Nachricht (10) Die Nachricht (10) von S nach M hatte zunächst den Inhalt $\{K_{CM}\}_{K_{MS}}$. Da diese Nachricht keinen Frischebeweis hat, war das **Ziel 5** nicht erreichbar. Erst durch das Einbinden des Werts N_M konnte M feststellen, dass auch K_{CM} frisch ist, was letztlich doch zum **Ziel 5** führt.

Doppelte Verschlüsselung in Nachricht (06) Zwischenzeitlich hatte die Nachricht (06), die von S nach M gesendet wird, den Inhalt $\{\{N_{S2}, N_C, ID_C\}_{K_{CS}}, \{N_{S2}\}_{K_{CS}}\}_{K_{MS}}$. Diese zusätzliche Verschlüsselung mit K_{MS} war aber unnötig, weil M den Schlüssel K_{CS} nicht kennt. Das Ableiten nach Protokollschritt (06) führt nur in eine Sackgasse. Dasselbe Ergebnis kann auch ohne die Verschlüsselung mit K_{MS} erreicht werden.

6 Fazit und Ausblick

Die Arbeit hat gezeigt, dass die entworfene Sicherheitsarchitektur die Anforderung aus dem Beispielszenario (siehe Kapitel 3.2) erfüllt. Die Schutzziele Datenintegrität und Informationsvertraulichkeit werden durch die Verwendung der RSN-Sicherheitsarchitektur gewährleistet. Für das Schutzziel Authentizität wurde das EAP-Verfahren EAP-CHIP entworfen. EAP-CHIP führt eine beidseitige Authentizitätsprüfung sowohl zwischen mobilem Gerät und Authentication-Server als auch zwischen Chipkarte und Authentication-Server durch. Zusätzlich beinhaltet EAP-CHIP die Prüfung der Authentizität des Benutzers durch eine PIN-Abfrage. Damit wird sichergestellt, dass ein Benutzer nur Zugang zum Wireless-LAN erhält, wenn dieser ein authentifizierbares Gerät und eine authentifizierbare Chipkarte hat und die PIN der Chipkarte kennt.

Neben dem in der Arbeit verwendeten Beispielszenario sind auch weitere Einsatzmöglichkeiten für die Sicherheitsarchitektur vorstellbar.

Beispielsweise kann ein Ferienclub seinen Gästen die Möglichkeit bieten, über mobile Geräte Informationen abzurufen und Bestellungen aufzugeben. Mit dieser Sicherheitsarchitektur können nur mobile Geräte, die vom Ferienclub authentifiziert sind, verwendet werden. Zusätzlich müssen diese mobilen Geräte nicht dem einzelnen Gast zugewiesen werden und jeder Gast kann jedes dieser Geräte verwenden. Sinnvoll ist die Sicherheitsarchitektur besonders dann, wenn der Gast bereits eine Chipkarte hat, mit der er z. B. seine Apartmenttür öffnen kann.

Vorstellbar ist auch die Verwendung der Sicherheitsarchitektur bei Sitzungen von unterschiedlichen Gremien (z. B. Senat, Gemeinderat, etc.). Jeder Teilnehmer erhält eine persönliche Chipkarte. Auf der jeweiligen Sitzung kann er mit seiner Chipkarte und einem für die Sitzung vorgesehenen mobilen Gerät Informationen zur Sitzung erhalten und persönliche Einträge und Kommentare eintragen.

Neben der gesamten Sicherheitsarchitektur kann auch EAP-CHIP unabhängig vom Wireless-LAN weiter verwendet werden. Da z. B. durch den Standard IEEE 802.1X EAP-Verfahren unterstützt werden, ist es möglich, EAP-CHIP auch in anderen Netzwerken einzusetzen.

Literaturverzeichnis

Aboba u. a. 2003 ABOBA, B. ; MICROSOFT ; CALHOUN, P. ; AIRESPACE: *RADIUS (Remote Authentication Dial In User Service) - Support For Extensible Authentication Protocol (EAP)*. 2003. – URL ftp://ftp.rfc-editor.org/in-notes/rfc3579.txt. – Zugriffsdatum: 2004-05-26

Beutelspacher u. a. 2004 BEUTELSPACHER, Albrecht ; SCHWENK, Jörg ; WOLFENSTETTER, Klaus-Dieter: *Moderne Verfahren der Kryptographie*. 5. Auflage. vieweg, 2004. – ISBN 3-528-46590-5

Blunk u. a. 1998 BLUNK, L. ; VOLLBRECHT, J. ; MERIT NETWORK, Inc.: *PPP Extensible Authication Protocol (EAP)*. 1998. – URL ftp://ftp.rfc-editor.org/in-notes/rfc2284.txt. – Zugriffsdatum: 2004-05-26

BSI 2002 : *Sicherheit im Funk-LAN (WLAN, IEEE 802.11)*. 2002. – URL http://www.bsi.bund.de/literat/doc/wlan/wlan.pdf. – Zugriffsdatum: 2004-02-19

Burrows u. a. 1990 BURROWS, Michael ; ABADI, Martin ; NEEDHAM, Roger: *A Logic of Authentication*. 1990. – URL http://www.citeseer.ist.psu.edu/burrows90logic.html. – Zugriffsdatum: 2004-04-29

Dworkin 2001 DWORKIN, Morris: *Recommendation for Block Cipher Modes of Operation - Methods and Techniques*. 2001. – URL http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf. – Zugriffsdatum: 2004-06-02

Eckert 2003 ECKERT, Claudia: *IT-Sicherheit Konzepte - Verfahren - Protokolle*. 2. überarbeitete und erweiterte Auflage. Oldenbourg, 2003. – ISBN 3-486-27205-5

Feistel 1973 FEISTEL, H.: Cryptography and Computer Privacy. In: *Scientific American* 228 (1973), Nr. 5

FIPS PUB 197 2001 : *Specification for the ADVANCED ENCRYPTION STANDARD (AES)*. 2001. – URL http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. – Zugriffsdatum: 2004-06-02

Grogans u. a. 2000 GROGANS, Candance ; BETHEA, Jackie ; HAMDAN, Issam: *RC4 Encryption Algorithm - Class Project: COMP-696 - Description: Covers the history of the RC4 algorithm and some details about the structure behind its design.* 2000. – URL `http://www.ncat.edu/~grogans/main.htm`. – Zugriffsdatum: 2004-05-24

H. Haverinen u. a. 2004 H. HAVERINEN, Ed. ; NOKIA ; J. SALOWEY, Ed. ; SYSTEMS, Cisco: *Extensible Authentication Protocol Method for GSM Subscriber Identity Modules (EAP-SIM)*. 2004. – URL `http://www.ietf.org/internet-drafts/draft-haverinen-pppext-eap-sim-13.txt`. – Zugriffsdatum: 2004-06-08

IANA 2004 : *Extensible Authentication Protocol (EAP) Registry*. 2004. – URL `http://www.iana.org/assignments/eap-numbers`. – Zugriffsdatum: 2004-05-26

IEEE 1999 : *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. 1999. – URL `http://standards.ieee.org/getieee802/download/802.11-1999.pdf`. – Zugriffsdatum: 2004-02-18

IEEE 2001 : *IEEE Standard for Local and metropolitan area networks - Port-Based Network Access Control*. 2001. – URL `http://standards.ieee.org/getieee802/download/802.1X-2001.pdf`. – Zugriffsdatum: 2004-05-26

IEEE 2003 : *Draft Amendment to STANDARD FOR Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements*. 2003

Kauffels 2002 KAUFFELS, Franz-Joachim: *Wireless LANs*. 1. Auflage. mitp, 2002. – ISBN 3-8266-0955-7

Meadows 1995 MEADOWS, Catherine: Formal Verification of Cryptographic Protocols: A Survey. In: *Proceedings of the 4th International Conference on the Theory and Applications of Cryptology*, Springer-Verlag, 1995, S. 135–150. – ISBN 3-540-59339-X

Rankl und Effing 2002 RANKL, Wolfgang ; EFFING, Wolfgang: *Handbuch der Chipkarten*. 4. überarbeitete und aktualisierte Auflage. Hanser, 2002. – ISBN 3-446-22036-4

Rigney u. a. 1997 RIGNEY, C. ; LIVINGSTON ; RUBENS, A. ; MERIT ; SIMPSON, W. ; DAYDREAMER ; WILLENS, S. ; LIVINGSTON: *Remote Authentication Dial In User Service (RADIUS)*. 1997. – URL `ftp://ftp.rfc-editor.org/in-notes/rfc2058.txt`. – Zugriffsdatum: 2004-05-26

Schneider 1996 SCHNEIDER, Bruce: *Angewandte Kryptographie: Protokolle Algorithmen und Sourcecode in C*. Addison-Wesley, 1996. – ISBN 3-89319-854-7

Whiting u. a. 2003 WHITING, D. ; HIFN ; HOUSLEY, R. ; SECURITY, Vigil ; FERGUSON, N. ; MACFERGUS: *Counter with CBC-MAC (CCM)*. 2003. – URL `ftp://ftp.rfc-editor.org/in-notes/rfc3610.txt`. – Zugriffsdatum: 2004-06-02

Zorn und Corporation 1999 ZORN, G. ; CORPORATION, Microsoft: *Microsoft Vendor-specific RADIUS Attributes*. 1999. – URL `ftp://ftp.rfc-editor.org/in-notes/rfc2548.txt`. – Zugriffsdatum: 2004-05-27

Abkürzungsverzeichnis

A2	address 2
AAD	Advanced Encryption Standard
AES	Advanced Encryption Standard
AP	Access-Point
AS	Authentication-Server
BAN	Burrows, Abadi und Needham
BSI	Bundesamt für Sicherheit in der Informationstechnik
BSS	Basic Service Set
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCM	Counter with CBC-MAC
CCMP	Counter-Mode/CBC-MAC Protocol
CCS	cryptographic checksum
CHAP	Challenge Handshake Authentication Protocol
CHV	card holder verification
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CTR	Counter Mode
DA	destination address
DS	Distribution System
EAP	Extensible Authentication Protocol
EAPOL	EAP over LAN
ESS	Extended Service Set
GSM	Global System for Mobile Communication
IANA	Internet Assigned Numbers Authority
IBSS	Independent Basic Service Set
ICV	Integrity Check Value
IEC	International Electrotechnical Commission

IEEE ..... Institute of Electrical and Electronics Engineers
IMSI ..... International Mobile Subscriber Identity
ISO ..... International Standardisation Organization
IV ..... Initialisierungsvektor
LAN ..... Local Area Network
MAC ..... Medium Access Control
MIC ..... Message Integrity Code
MPDU ..... MAC protocol data unit
MPPE ..... Microsoft Point-to-Point Encryption
MSDU ..... MAC service data unit
PAP ..... Password Authentication Protocol
PIN ..... personal identification number
PMK ..... Pairwise Master Key
PN ..... Paketnummer
POS ..... Point of Sell
PPP ..... Point-to-Point Protocols
PRF ..... Pseudo-Random Function
PTK ..... Pairwise Transient Key
PTT ..... Post, Telegraph and Telephone Administration
RADIUS ..... Remote Authentication Dial In User Service
RC4 ..... Rons Code 4
RSN ..... Robust Security Network
SA ..... source address
SIM ..... GSM Subscriber Identity Module
STA ..... Station
TA ..... transmitter address
TKIP ..... Temporal Key Integrity Protocol
TSC ..... TKIP sequence counter
TTAK ..... TKIP mixed Transmit Address and Key
WEP ..... Wired Equivalent Privacy
WEP IV ..... WEP Initialisierungsvektor

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

30. Juni 2004	
Ort, Datum	Unterschrift