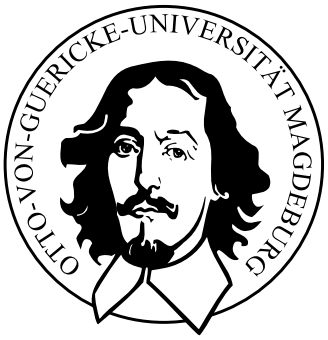

Fakultät für Informatik

Institut für Verteilte Systeme

Diplomarbeit

Security in drahtlosen Mesh-Netzwerken

Christian Fackroth

16. September 2008

Betreuer:
Prof. Dr. Edgar Nett
Dipl.-Inform. Georg Lukas

Fackroth, Christian:
Security in drahtlosen Mesh-Netzwerken
Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 2008.

Danksagung

Ich möchte diese Gelegenheit nutzen, mich bei den zahlreichen Personen zu bedanken, die mich während der Entstehung dieser Arbeit durch fachliche Hilfe, Ratschläge und auch durch Motivation unterstützt haben.

Einen großen Dank gilt den Professoren Edgar Nett und Jörg Kaiser für ihre Tätigkeit als Gutachter dieser Arbeit. Weiterhin möchte ich mich bei meinem Betreuer Georg Lukas bedanken, der mir in zahlreichen Gesprächen und Diskussionen half den richtigen Weg für die Anfertigung dieser Arbeit einzuschlagen.

Auch möchte ich mich bei meiner Familie bedanken, die mir während des gesamten Studiums Rückhalt geboten hat. Schließlich gilt mein Dank noch allen Freunden, die mich bei der Entstehung dieser Arbeit moralisch, wie auch konstruktiv unterstützt haben.

Kurzfassung

Diese Arbeit beschäftigt sich mit der Entwicklung von Verfahren, zum Schutz drahtloser Mesh-Netzwerke vor unberechtigter Informationsgewinnung Dritter. Dabei war es ein Ziel, den vorhandenen IEEE 802.11i Standard für Ad-hoc-Netzwerke in einer prototypischen Implementierung zu realisieren, so dass ein Security-Verfahren für drahtlose Mesh-Netzwerke zur Verfügung gestellt werden kann, welches ein hohes Maß an Security bietet. Dies beinhaltet die Wahrung der Vertraulichkeit und Integrität der Daten, sowie eine Authentifizierung zwischen den Stationen. Da dieses Verfahren jedoch für die Bedürfnisse von Ad-hoc-Netzwerken konzipiert ist, zieht die Verwendung des Verfahrens Einbußen in der Performance, sowie Einschränkungen in der Flexibilität von drahtlosen Mesh-Netzwerken nach sich. Daher bestand ein weiteres Ziel darin, auf der Grundlage des vorhandenen IEEE 802.11i Standards ein weiteres Verfahren zu entwickeln, welches einen besseren Kompromiss, zwischen den Bedürfnissen von drahtlosen Mesh-Netzwerken und Maßnahmen zur Verhinderung unberechtigter Informationsgewinnung Dritter, erzielt.

Da der IEEE 802.11i Standard für Ad-hoc-Netzwerke nur wesentliche Eckpunkte definiert, wird zunächst ein detailliertes Konzept für eine Implementierung erarbeitet. Auf dieser Grundlage wird anschließend ein Konzept für ein neues, speziell auf die Bedürfnisse von drahtlosen Mesh-Netzwerken angepasstes, Verfahren entwickelt, welches nur geringfügige Abstriche in der Security macht, aber gleichzeitig die Leistungsfähigkeit von drahtlosen Mesh-Netzwerken, speziell bei der Nutzung mobiler Stationen, nahezu unverändert erhält. Anschließend werden die erarbeiteten Konzepte, unter Nutzung bestehender Software-Komponenten in prototypischen Implementierungen realisiert. Jedoch stellte sich die Implementierung der konzipierten Schlüsselverwaltung des IEEE 802.11i Standards für Ad-hoc-Netzwerke als recht aufwendig heraus, so dass an dieser Stelle eine Vereinfachung gegenüber dem Konzept vorgenommen wird. Dies führt dazu, dass die prototypische Implementierung, entgegen dem Konzept, Multicast-Verbindungen mit lediglich drei Nachbarstationen ermöglicht.

Abschließend werden die prototypischen Implementierungen, anhand von verschiedenen Szenarien auf ihre Funktionalität und Leistungsfähigkeit überprüft. Dabei wird ersichtlich, dass beide Implementierungen der erwarteten Leistungsfähigkeit gerecht werden, wobei die prototypische Implementierung des IEEE 802.11i Standards aufgrund der vereinfachten Implementierung und der Flexibilität eine eingeschränkte und das an den Bedürfnissen von drahtlosen Mesh-Netzwerken angepasste Verfahren eine uneingeschränkte praktische Verwendung erlauben.

Inhaltsverzeichnis

Verzeichnis der Abkürzungen und Akronyme

AES	Advanced Encryption Standard
AP	Access Point
CBC	Cipher Block Chaining
CCM	Counter Mode-CBC Message Authentication Code
CCMP	Counter Mode-CBC Message Authentication Code Protocol
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CTR	Counter Mode
EAP	Extensible Authentication Protocol
EAPOL	EAP Over LAN
FCS	Frame Check Sequence
GMK	Group Master Key
GTK	Group Transient Key
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific and Medical
ICV	Integrity Check Value
IV	Initialization Vector
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MIC	Message Integrity Code
OCB	Offset Codebook Mode
PMK	Pairwise Master Key
PN	Packet Number
PSK	Pre-Shared Key
PTK	Pairwise Transient Key
RADIUS	Remote Authentication Dial-In User Service
RSNA	Robust Security Network Association
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security

TSC	TKIP Sequence Counter
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access Version 2
WRAP	Wireless Robust Authentication Protocol

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

Die vorliegende Arbeit beschäftigt sich mit dem Aspekt der Security in drahtlosen Mesh-Netzwerken. Dieses Kapitel dient dazu, zunächst einen Überblick über die vorliegende Arbeit zu vermitteln. Einführend zur Problematik wird die Motivation, die zu dieser Arbeit führte, vorgestellt, woraus sich die im darauf folgenden Abschnitt geschilderte Aufgabenstellung ergibt.

Im dritten Abschnitt dieses Kapitels werden die Ergebnisse dieser Arbeit in knapper Form erläutert, bevor das Kapitel mit dem Aufbau der übrigen Arbeit abgeschlossen wird.

1.1 Motivation

Die Informationstechnologie nimmt eine essenzielle Bedeutung in der heutigen Gesellschaft ein. Der Drang ständig mit anderen Nachrichten auszutauschen und dabei unabhängig vom eigenen Standort zu sein, führte zur Entwicklung zahlreicher neuer Technologien. Zu ihnen gehören beispielsweise die Entwicklung von Handys, zur drahtlosen Telefonie, oder das von immer mehr Geräten unterstützte Wireless Local Area Network *(WLAN)*.

Besonders die WLAN-Technologie gewinnt zunehmend an Bedeutung, sowohl im Privaten als auch im Industriellen Umfeld. So können unabhängig vom Standort schnell und kostengünstig WLANs zur Kommunikation im Nahbereich eingerichtet werden. Dabei hat sich die Nutzung einer durch Access Points *(APs)* realisierten, mehr oder weniger aufwendigen Infrastruktur etabliert. Ein AP fungiert als zentraler Koordinator und stellt für einen lokal begrenzten Bereich meist einen Zugriffspunkt auf ein weiteres Netzwerk, beispielsweise das Internet, dar. Für solche Infrastruktur-Netzwerke, in denen der AP einen zentralen Zugriffspunkt darstellt, sind Mechanismen zum Schutz der übermittelten Informationen vor unberechtigtem Zugriff Dritter in ausgereifter Form verfügbar.

Soll über ein größeres Gebiet, beispielsweise in großen Industrieanlagen, ein WLAN eingerichtet werden, so müssen mehrere APs verwendet werden. Dabei ist es notwendig, dass die APs untereinander kommunizieren können. Die Kommunikation erfolgt dabei meist über eine Kabelanbindung. Damit verbunden sind Einschränkungen in der Nutzung eines solchen WLANs. So können diese WLANs in großen Industrieanlagen, wie es in der Prozessindustrie der Fall ist, oder in Katastrophen- und Krisengebieten nur mit erheblichen finanziellen und zeitlichem Aufwand realisiert werden. Im schlimmsten Fall, z.B. in denkmalgeschützten Gebäuden, ist eine Verkabelung gar unmöglich.

Die fortgeschrittene technologische Entwicklung im Bereich der WLANs erlaubt es nunmehr, mit den verfügbaren WLAN-Geräten, auch Netzwerke ohne Infrastruktur aufzubauen. So lassen sich einfach drahtlose Ad-hoc-Netzwerke mit geringem

Zeitaufwand und geringen Kosten realisieren. In einem drahtlosen Ad-hoc-Netzwerk kann ein WLAN-Gerät mit allen, in direkter Sende- und Empfangsreichweite befindlichen, WLAN-Geräten kommunizieren. Zur weiteren Erhöhung der Flexibilität, lassen sich auf Grundlage von drahtlosen Ad-hoc-Netzwerken drahtlose Mesh-Netzwerke *(WMNs)* aufbauen. In einem WMN bilden einige WLAN-Geräte ein stationäres drahtloses Netzwerk, welches sich analog zu Infrastruktur-Netzwerken an weitere Netzwerke anbinden lässt, während andere WLAN-Geräte innerhalb des Netzwerks mobil sind. Um eine Kommunikation auch zwischen WLAN-Geräten außerhalb der direkten Reichweite zu ermöglichen, können WLAN-Geräte als Zwischenstationen, zur Weiterleitung der Informationen, genutzt werden. So lassen sich kabellos, kostengünstig und mit geringem Zeitaufwand auch über größere Gebiete erstreckende dezentrale WLAN-Netzwerke etablieren.

Mit dem Wegfall des APs als zentrale Instanz in WMNs greifen jedoch die ausgereiften Mechanismen, zum Schutz vor unberechtigten Zugriff Dritter, aus Infrastruktur-Netzwerken nicht. Um auch in WMNs die übermittelten Informationen vor unberechtigten Dritten, in einem hohen Maß zu schützen und damit die Akzeptanz von Anwendern zu gewinnen, ist es notwendig dem Anwender ein entsprechendes Verfahren zur Verfügung zu stellen.

Für drahtlose Ad-hoc-Netzwerke sieht der IEEE[1] 802.11i Standard [IEE04] bereits ein Verfahren, durch Adaptierung der Mechanismen von Infrastruktur-Netzwerken, vor. Da WMNs auf drahtlose Ad-hoc-Netzwerke basieren, lassen sich diese Mechanismen des IEEE 802.11i Standards gleichermaßen für WMNs verwenden. Allerdings ist für drahtlose Ad-hoc-Netzwerke bisher keine Umsetzung dieses Standards verfügbar.

Da die Realisierung eines Schutzes, der übermittelten Informationen vor unberechtigten Dritten, immer einen erhöhten Ressourcenbedarf nach sich zieht und in WMNs andere Voraussetzungen für den Schutz der Informationen, im Vergleich zu drahtlosen Ad-hoc-Netzwerken, gelten, ist es zur Erhaltung der Vorzüge von WMNs wünschenswert, ein auf die Bedürfnisse von WMNs angepasstes Verfahren zur Verfügung zu haben. Dabei sollte dieses Verfahren einen möglichst hohen Schutz bieten, wozu in erster Linie die Gewährleistung der Vertraulichkeit der Informationen, während der Übertragung, gehört. Aber auch die Sicherstellung der Integrität der übertragenden Informationen und die Authentizität der kommunizierenden WLAN-Geräte sollten, neben der effizienten Nutzung der technologischen Ressourcen, Berücksichtigung finden.

1.2 Aufgabenstellung

Da ein hinreichender Schutz der übertragenden Informationen vor unberechtigten Dritten in WMNs mit der Umsetzung des IEEE 802.11i Standards für den Ad-hoc-Modus gewährleistet werden kann, jedoch keine Umsetzung verfügbar ist, ist es ein Ziel

[1] Institute of Electrical and Electronics Engineers

dieser Arbeit den IEEE 802.11i Standard für den Ad-hoc-Modus in prototypischer Form umzusetzen.

Diese prototypische Umsetzung des IEEE 802.11i Standards bildet die Grundlage dafür, ein spezielles Verfahren für WMNs zu entwickeln. Wie auch das auf den IEEE 802.11i Standard basierende Verfahren, soll es die Vertraulichkeit, Integrität und Authentizität gegenüber nicht authentifizierten WLAN-Geräten gewährleisten, sowie die Flexibilität von WMNs nicht einschränken.

Den Ausgangspunkt für die Umsetzung beider Verfahren bildet die Open-Source-Software *hostapd* [MaH08] und *wpa_supplicant* [MaS08] von Jouni Malinen. Diese implementieren den Pflichtteil des IEEE 802.11i Standards für Infrastruktur-Netzwerke, unterstützen eine Vielzahl an WLAN-Geräten und sind für diverse Betriebssysteme verfügbar. Die Implementierung erfolgt, analog zur bereits bestehenden Software, in der Programmiersprache C.

Abschließend sollen beide Verfahren mit geeigneten Mitteln evaluiert werden. Dazu soll der Einfluss der Security-Mechanismen auf die Performance von WMNs in verschiedenen Szenarien untersucht werden.

1.3 Ergebnisse

Im Rahmen dieser Arbeit konnte gezeigt werden, dass der IEEE 802.11i Standard für Ad-hoc-Netzwerke in einer Implementierung umgesetzt werden kann und somit eine Möglichkeit darstellt, WMNs vor den Zugriff unberechtigter Dritter, zu schützen. Bei der Umsetzung stellte sich allerdings heraus, dass einige Forderungen des IEEE 802.11i Standards für Ad-hoc-Netzwerke in WMNs zum einen, wenig praktikabel sind und so zu Einschränkungen in praktischen Szenarien führen und zum anderen, dass im Vergleich zu Infrastruktur-Netzwerken ein hoher Kommunikationsaufwand zur Authentifizierung, sowie ein hoher Ressourcenbedarf zur Schlüsselverwaltung einkalkuliert werden muss. Der Ressourcenbedarf zur Umsetzung der Schlüsselverwaltung führte dazu, dass die prototypische Implementierung des IEEE 802.11i Standards für den Ad-hoc-Modus eine Multicast-Verbindung, lediglich zu drei WLAN-Geräten in der Nachbarschaft, ermöglicht.

Weiterhin konnte gezeigt werden, dass mit den Security-Mechanismen, die der IEEE 802.11i Standard zur Verfügung stellt, ein effizientes Security-Verfahren speziell für WMNs zur Verfügung gestellt werden kann. Dieses erhält die Flexibilität von WMNs, indem beispielsweise ein Roaming, verursacht durch das verwendete Security-Verfahren, entfällt. Darüber hinaus bietet es ebenso für reale Anwendungsszenarien ein hohes Maß an Security. Es kann mit beliebig vielen WLAN-Geräten verwendet werden und setzt in der prototypischen Implementierung lediglich eine geringfügige Anpassung, im Verschlüsselungsprotokoll, des Treibers voraus.

Bei der Untersuchung der beiden entwickelten Verfahren hat sich gezeigt, dass beide prototypischen Implementierungen erwartungsgemäß funktionsfähig sind und darüber

hinaus den erwarteten Performancewerten, weitestgehend gerecht wurden. Aufgrund der erzielten Performancewerte, der Erhaltung der Flexibilität, den nur geringfügig benötigten Anpassungen am verwendeten Treiber, sowie der eingeschränkten Möglichkeit der Schlüsselverwaltung der prototypischen Implementierung des IEEE 802.11i Standards für Ad-hoc-Netzwerke, stellt das an die Bedürfnisse von WMNs angepasste Verfahren das praktikablere Security-Verfahren für WMNs dar, welches beispielsweise in der Prozessindustrie seinen Einsatz finden könnte.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in sechs Kapitel. Nachdem in diesem Kapitel die Beweggründe, die zu dieser Arbeit geführt haben und die sich daraus ergebende Aufgabenstellung, gefolgt von einer Zusammenfassung der Ergebnisse geschildert wurden, wird im kommenden Kapitel auf die Grundlagen für diese Arbeit eingegangen. Dazu werden zunächst drahtlose Mesh-Netzwerke und die Problematik der Security eingeführt. Aufbauend darauf wird der Kontext zwischen Security und WLANs hergestellt, bevor das zweite Kapitel mit der Vorstellung der bereits verfügbaren Applikationen abgeschlossen wird.

Im dritten Kapitel wird das Konzept, zur Realisierung des IEEE 802.11i Standards für drahtlose Ad-hoc-Netzwerke, sowie ein darauf aufbauendes, speziell für die Bedürfnisse von WMNs entwickeltes, Verfahren vorgestellt. Zusammenfassend werden schließlich beide Verfahren einem Vergleich unterzogen.

Das vierte Kapitel befasst sich mit den Implementierungsspezifischen Gesichtspunkten der Konzepte aus Kapitel drei. Dabei wird ebenfalls eine Unterscheidung zwischen dem IEEE 802.11i Standard für den Ad-hoc-Modus und dem für WMN angepassten Verfahren vorgenommen.

Kapitel fünf beschäftigt sich mit der Evaluierung der entwickelten Verfahren. Die Evaluierung umfasst eine Überprüfung der Funktionstüchtigkeit der implementierten Verfahren, sowie eine Untersuchung hinsichtlich ihrer Skalierbarkeit. Anschließend werden die Ergebnisse der Untersuchungen beider Verfahren miteinander verglichen, bevor das Kapitel mit einer Überprüfung des Mechanismus zur Schlüsselaktualisierung, des an WMN angepassten Verfahrens, abgeschlossen wird.

Um die vorliegende Arbeit schließlich abzurunden, werden in Kapitel sechs die Ergebnisse zusammengefasst und ein Ausblick auf mögliche zukünftige Arbeiten gegeben.

2 Grundlagen und verwandte Arbeiten

Dieses Kapitel dient dazu, die für die Arbeit notwendigen Grundlagen zu vermitteln. Dazu wird zunächst darauf eingegangen was drahtlose Mesh-Netzwerke sind und wie sie sich von gewöhnlichen Rechnernetzen unterscheiden. Danach wird der Begriff der Security näher erläutert und eine Einteilung von kryptographischen Methoden vorgenommen. Aufbauend darauf werden etablierte Security-Mechanismen für WLANs vorgestellt, gefolgt von einigen bekannten Angriffen auf ihnen. Das Kapitel wird schließlich mit der Vorstellung der vorhandenen Software, die als Basis für diese Arbeit dient, abgeschlossen.

2.1 WLANs und drahtlose Mesh-Netzwerke

Das Institute of Electrical and Electronics Engineers *(IEEE)* legt eine Reihe von Standards für Local Area Networks *(LANs)* fest. Darunter auch den IEEE 802.11 Standard [IEE99, IEE07], welcher die drahtlose Kommunikation in WLANs auf der physikalischen Ebene und auf der Zugriffskontrollebene *(MAC[1]-Ebene)* regelt. Wie bei allen Standardisierungen trug auch der IEEE 802.11 Standard dazu bei, dass dem Anwender inzwischen eine große Auswahl kostengünstiger Endgeräte, unterschiedlicher Hersteller, zur Verfügung steht. Die Geräte sind untereinander kompatibel und können dazu genutzt werden, Netzwerke unter Verwendung einer Basisstation, dem AP, mit einer Infrastruktur *(Infrastruktur-Netzwerke)*, wie auch Netzwerke ohne Basisstation *(Ad-hoc-Netzwerke)* aufzubauen. Dabei bildet der räumliche Bereich innerhalb dem die Geräte, unter Berücksichtigung ihrer Sende- und Empfangsreichweite, untereinander kommunizieren können, eine *Zelle*.

Wie auch andere Netzwerk-Protokolle, lässt sich der IEEE 802.11 Standard in das allgemeine ISO/OSI-Referenzmodell eingliedern. Dabei entspricht die physikalische Ebene des IEEE 802.11 Standards genau der Schicht eins *(Bitübertragungsschicht)* des ISO/OSI-Referenzmodells, während die MAC-Ebene nur einen Teil der Schicht zwei *(Sicherungsschicht)* des ISO/OSI-Referenzmodells beschreibt.

Die Datenübertragung auf der physikalischen Ebene erfolgt auf zwei zur Verfügung stehenden Funkfrequenzbänder, unter Verwendung unterschiedlicher Modulationsverfahren [NMG01]. Auf dem ISM-Frequenzband *(Industrial, Scientific and Medical-Band)* stehen in Europa 13 Kanäle und in den USA 11 Kanäle, im Bereich zwischen 2400 MHz und 2483,5 MHz, zur Verfügung. Diese können in geschlossenen Gebäuden, wie auch außerhalb, ohne Restriktionen lizenzfrei genutzt werden und erlauben Brutto-Datenraten von bis zu 54 Mbit/s [IEE03g, IEE07].

Das zweite Frequenzband liegt im 5-GHz-Band von 5150 MHz bis 5725 MHz und wurde erst Ende 2002 in Deutschland freigegeben. Trotz der Freigabe unterliegt es einigen Restriktionen, so darf es beispielsweise außerhalb geschlossener Gebäude nur

[1] Media Access Control

mit Implementierung des IEEE 802.11h Standards [IEE03h, IEE07] genutzt werden. Dieser schreibt zusätzliche Mechanismen, wie die automatische Absenkung der Sendeleistung und einen automatischen Frequenzwechsel bei Erkennung fremder Geräte vor. Auch hier werden Brutto-Datenraten von bis zu 54 Mbit/s erreicht [IEE99a, IEE07]. Allerdings sorgen die genannten Restriktionen dafür, dass die Technik verhältnismäßig teuer ist und daher nur eine allmähliche Verbreitung erfährt.

Unmittelbar oberhalb der physikalischen Ebene liegt die MAC-Ebene. Diese übernimmt die aus LANs bekannten Aufgaben der Adressierung, den Aufbau von Nachrichten in Kopf- und Datenteil, sowie die Fragmentierung langer Datenpakete[1] in die, für die Versendung notwendigen Frames[2]. Somit ist ab dieser Ebene eine Kompatibilität mit dem etablierten IEEE 802.3 Standard gewährleistet. Allerdings hat die MAC-Ebene darüber hinaus noch weitere Aufgaben zu erfüllen [NMG01].

- Umsetzung von Security-Mechanismen
- Umsetzung von Roaming-Mechanismen
- Umsetzung von Stromsparmechanismen
- Regelung eines kollisionsfreien Zugriffs *(CSMA/CA[3]-Verfahren [FG97])*
- Aufbau und Unterscheidung zwischen Infrastruktur- und Ad-hoc-Netzwerken

Im Gegensatz zu einem Infrastruktur-Netzwerk, gibt es in einem Ad-hoc-Netzwerk keinen zentralen Koordinator, in Form eines APs. Vielmehr kann jede *Station* einer Zelle mit jeder anderen, sich in Reichweite befindenden, Station der Zelle direkt kommunizieren. Da außer den Stationen keine weitere Infrastruktur notwendig ist, ist die Konfiguration und das Betreiben von Ad-hoc-Netzwerken einfach und kostengünstig. Allerdings haben Ad-hoc-Netzwerke die Nachteile, dass sie nur über eine beschränkte Reichweite verfügen und darüber hinaus eine Anbindung an weitere Netzwerke, beispielsweise dem Internet nicht ermöglichen. Diese Nachteile lassen sich durch die Verwendung eines drahtlosen Mesh-Netzwerks *(WMNs)*, auf Grundlage des Ad-hoc-Modus des IEEE 802.11 Standards, beheben. Um eine Verbindung zu einem weiteren Netzwerk, sowohl drahtgebunden als auch kabellos zu ermöglichen, wie es in Infrastruktur-Netzwerken mittels des zentralen APs möglich ist, nehmen einige Stationen eine feste Position ein. Die übrigen Stationen behalten ihre Mobilität, analog den Stationen in einem gewöhnlichen Ad-hoc-Netzwerk bei. Der weitere Nachteil, der beschränkten Reichweite, aufgrund der begrenzten Sendeleistung einer einzelnen Station, wird durch die Einführung eines Multihop-Routing-Protokolls, beispielsweise AWDS[4] [AWDS08] gelöst. Das Multihop-Routing-Protokoll ermöglicht die Kommunikation mit jeder Station im Netzwerk, indem andere Stationen *(Router-Stationen)* dazu verwendet werden, die Informationen weiter zu vermitteln. So können Änderungen in der Topologie des Netzwerkes, beispielsweise durch Mobilität oder

1 Auch als MAC Service Data Units *(MSDU)* bezeichnet
2 Auch als MAC Protocol Data Units *(MPDU)* bezeichnet
3 Carrier Sense Multiple Access / Collision Avoidance
4 Ad-hoc Wireless Distribution System

etwaigen Hindernissen, durch entsprechendes Routing in einem Gewissen Umfang kompensiert werden. Gleichermaßen wird durch diese Maßnahmen eine hohe Flexibilität, eine Reduzierung der Kosten durch Vermeidung von Verkabelungen und ein breites Anwendungsspektrum [AWW04] erzielt. Beispiele für den Einsatz von WMNs sind: die Prozessindustrie, die Verwendung in Katastrophen- und Krisengebiete, da dort meist keine Infrastruktur vorhanden ist, sowie in Gebäuden wo keine Kabel verlegt werden können, z.B. in Denkmalgeschützten Gebäuden.

2.2 Einführung in die Security

2.2.1 Allgemeines

In der englischen Sprache gibt es zwei Begriffe *Safety* und *Security*, deren deutsche Übersetzung gleichermaßen Sicherheit ist. Allerdings grenzen beide Begriffe zwei unterschiedliche Bereiche der Sicherheit näher ein. Der Begriff Safety entspricht der Funktions- bzw. der Betriebssicherheit [Tan03] eines Systems. Um Safety zu gewährleisten, darf ein System keine unzulässigen Zustände annehmen können. Unter Security hingegen wird die Angriffssicherheit bzw. die IT-Sicherheit [Eck04] eines Systems verstanden. Dies beinhaltet Maßnahmen im System, die eine unautorisierte Veränderung oder Gewinnung von Informationen verhindern sollen.

Um den für diese Arbeit relevanten Bereich der Sicherheit näher einzugrenzen, wird der internationale Fachbegriff Security in dieser Arbeit verwendet. Nachfolgend wird auf die fünf allgemeinen Aspekte [Eck04] der Security näher eingegangen.

Authentizität

Unter Authentizität[1] wird der Nachweis der Identität des Urhebers, der Nachweis der Originalität des Datenmaterials oder einer Kommunikationsbeziehung verstanden. Um die Authentizität nachzuweisen, muss eine Authentifizierung[2] vorgenommen werden. Dies kann beispielsweise über eine Passwortabfrage, biometrische Merkmale, durch digitale Signaturen oder Zertifikate erfolgen.

Integrität

Unter Integrität[3] wird die Unversehrtheit der gespeicherten und übertragenen Daten verstanden. Dabei muss gewährleistet werden, dass keine Veränderungen durch unautorisierte Dritte, unerkannt bleiben. Die genutzten Maßnahmen ermöglichen jedoch nur die Erkennung von Manipulationen und keine Wiederherstellung der Originalinformationen.

---

1 Engl.: Authenticity
2 Engl.: Authentication
3 Engl.: Integrity

Vertraulichkeit

Durch Vertraulichkeit[1] wird verhindert, dass unautorisierte Dritte auf die Daten zugreifen können. Dies kann so strikt realisiert werden, dass selbst das Wissen über die Existenz der Daten vertraulich ist.

Verfügbarkeit

Verfügbarkeit[2] im Sinne der Security ist die Gewährleistung, dass ein Dienst authentifizierten Parteien zu jedem Zeitpunkt zur Verfügung steht und nicht durch unautorisierte Dritte beeinträchtigt werden kann.

Verbindlichkeit

Unter Verbindlichkeit[3] wird eine Möglichkeit verstanden, dass neben dem Datenempfänger auch berechtigte Dritte Authentizität und Integrität der Daten prüfen können. So ist es dem Urheber unmöglich, die Durchführung der Aktion zu leugnen.

Kryptographie

Das Ziel, aus Sicht der Security, bei der Übertragung von Nachrichten in Netzwerken ist es, die zu übertragenden Nachrichten vor unberechtigten Dritten zu schützen. Dies beinhaltet die Gewährleistung der Authentizität, Integrität und Vertraulichkeit der Nachrichten, was mit Hilfe eines kryptographischen Verfahrens realisiert werden kann. Kryptographische Verfahren sorgen dafür, dass die ursprüngliche Nachricht *(Klartext)* mit Hilfe einer Transformation in eine andere, neue Nachricht *(Chiffretext)*, überführt wird. Der resultierende Chiffretext soll dabei möglichst keinen Zusammenhang zum Klartext aufweisen. Damit Qualitätskontrollen und Standardisierungen der Algorithmen einfacher möglich sind, sind die Algorithmen meist bekannt. Um dennoch die Nachrichten individuell schützen zu können, erhält der Algorithmus neben dem Klartext einen zusätzlichen Parameter. Dieser Parameter *(Schlüssel)* sorgt dafür, dass nur authentisierte Parteien, Nachrichten Ver- und Entschlüsseln können.

2.2.2 Symmetrische Kryptographie

Symmetrische Kryptographie ist der klassische Weg Nachrichten bei der Übertragung vor unbefugten Zugriff zu schützen. Die dahinter stehende Idee ist die, dass die befugten Parteien sich auf die Schlüssel zum Verschlüsseln *(Chiffrierschlüssel)* und Entschlüsseln *(Dechiffrierschlüssel)* einigen. Diese werden dann für die Verschlüsselung und Entschlüsselung, der zu übertragenden Nachrichten genutzt. Dabei

[1] Engl.: Confidentiality
[2] Engl.: Availability
[3] Engl.: Non-Repudiation

gilt, dass der Chiffrierschlüssel sich aus dem Dechiffrierschlüssel und umgekehrt berechnen lässt. Bei der Mehrheit der Algorithmen sind Chiffrier- und Dechiffrierschlüssel der Einfachheit halber gar identisch.

Eine Übertragung von Nachrichten würde, beispielsweise zwischen den zwei Parteien Alice und Bob, folgendermaßen ablaufen. Zunächst müssen sich Alice und Bob auf einen Verschlüsselungsalgorithmus verständigen. Nun ist es notwendig einen Schlüssel zur Kommunikation zu vereinbaren. Sind diese zwei Schritte erledigt, können Alice und Bob ihre Nachrichten vor unbefugten Zugriff, während der Übertragung, schützen. Dazu erzeugt Alice, bei der Verwendung eines *Stromchiffrierungsalgorithmus*, einen Zufallsstrom, mit dem geheimen Schlüssel und verknüpft diesen XOR[1] mit ihrer Nachricht oder verschlüsselt die Nachricht, bei der Verwendung eines *Blockchiffrierungsalgorithmus*, direkt mit Hilfe des geheimen Schlüssels, bevor sie die Nachricht an Bob sendet. Bob kann schließlich seinerseits die Nachricht mit dem vereinbarten Verschlüsselungsalgorithmus und dem geheimen Schlüssel entschlüsseln.

Blockchiffrierung und Stromchiffrierung

Algorithmen der symmetrischen Kryptographie lassen sich in zwei unterschiedliche Gruppen einteilen. Die erste Gruppe bilden die *Blockchiffrierungsalgorithmen*. Hier wird der Klartext bei der Verschlüsselung blockweise verarbeitet, so dass für einen Block Klartext genau ein Block Chiffretext, gleicher Länge, entsteht. Bei der Entschlüsselung wird der Chiffretext entsprechend blockweise verarbeitet. Die Blocklänge beträgt heute üblicherweise 64 Bit oder 128 Bit, kann sich aber auch abhängig vom verwendeten Verfahren davon unterscheiden. Die zweite Gruppe stellen die *Stromchiffrierungsalgorithmen* dar. Hier wird der Klartext meist Bitweise oder Byteweise verarbeitet, indem der durch den Stromchiffrierungsalgorithmus erzeugte Zufallsstrom XOR mit dem Klartext verknüpft wird. Dies hat den Vorteil, dass das Verfahren unabhängig von der Länge des Klartextes bzw. Chiffretextes ist.

Wird bei einem Blockchiffrierungsalgorithmus der letzte Block nicht komplett gefüllt, muss dieser aufgefüllt werden. Dieser Vorgang, der als Padding bezeichnet wird, füllt den Rest des Blockes mit einer eins, gefolgt von Nullen auf, um so bei der Entschlüsselung die ungenutzten Daten leichter entfernen zu können.

Ein Blockchiffrierungsalgorithmus überführt dabei unter Verwendung des gleichen Schlüssels ein und denselben Klartextblock in den gleichen Chiffretextblock. Dies ist bei einem Stromchiffrierungsalgorithmus nicht der Fall. Hier entsteht aus dem gleichen Klartextbit oder Klartextbyte, bei jeder Verschlüsselung ein anderes. Allerdings gibt es die Möglichkeit die einzelnen Blöcke der Blockchiffrierungsalgorithmen auf unterschiedliche Weise miteinander zu verknüpfen. Der Vorgang der Verknüpfung wird als kryptographischer Modus bezeichnet und kann unter anderem dazu genutzt werden Blockchiffrierungsalgorithmen in Stromchiffrierungsalgorithmen zu überführen. Der

[1] XOR = Exclusive Or

umgekehrte Weg ist nicht möglich, was zeigt, dass Blockchiffrierungsalgorithmen allgemeiner und damit auch flexibler eingesetzt werden können. Da die Sicherheit durch den eingesetzten Algorithmus gewährleistet werden soll, ist der Vorgang der Verknüpfung meist einfach und damit effizient umgesetzt.

Nachfolgend werden nun einige kryptographische Modi aufgezeigt, die für den Kontext der vorliegenden Arbeit relevant sind. Weiterführende Modi, sowie detailliertere Informationen zu den Anforderungen, die an kryptographische Modi gestellt werden können [Sch96] entnommen werden.

Cipher Block Chaining

Beim Cipher Block Chaining *(CBC)* hängt der resultierende Chiffreblock nicht nur von dem zugrunde liegenden Klartextblock ab, sondern zusätzlich von allen vorherigen Klartextblöcken. Dazu wird bei der Verschlüsselung jeder zu verschlüsselnde Klartextblock mit dem vorherigen Chiffreblock XOR verknüpft, bis zur Verschlüsselung der kompletten Nachricht. Je nach Verwendung kann es dabei einen oder zwei Sonderfälle geben.

Der erste Sonderfall ist der erste Klartextblock, da noch kein Chiffreblock existiert. Eine naheliegende Möglichkeit ist es, den ersten Klartextblock normal, ohne eine XOR-Verknüpfung, zu verschlüsseln. Das Problem was sich allerdings daraus ergibt ist, dass zwei Nachrichten deren Klartextanfang gleich ist, z.B. der Adresskopf einer E-Mail, auch den gleichen Chiffretextanfang hätten. Daraus könnte ein Kryptoanalytiker[1] nützliche Informationen erhalten, was vermieden werden sollte. Die Lösung für dieses Problem ist die Erzeugung eines ersten Chiffreblocks aus Zufallsdaten. Diese Zufallsdaten werden als *Initialisierungsvektor (IV)* bezeichnet und Bewirken, dass identische Nachrichten verschieden verschlüsselt werden. Es ist dabei nicht notwendig, dass der IV geheim bleibt, da für jede Verschlüsselung eines Klartextblockes der vorherige Chiffreblock als IV fungiert. So ergeben sich für *n* Chiffreblöcke *n-1* IVs, die bekannt sein müssten.

Der zweite Sonderfall ist der letzte Block. Dieser kann entweder, wie bereits beschrieben, aufgefüllt werden oder falls die Nachricht ihre länge beibehalten muss, gesondert verschlüsselt werden. Dazu wird der letzte vollständige Chiffreblock zunächst ein weiteres Mal verschlüsselt und anschließend werden die ersten *n*-Bits des entstandenen Resultates, entsprechend der Länge des zu verschlüsselnden Klartextes, XOR mit dem Klartext verknüpft.

[1] Person, die sich mit dem brechen von Chiffretext beschäftigt

Counter Mode

Beim Counter Mode *(CTR)* werden Sequenznummern mit Hilfe eines Blockalgorithmus verschlüsselt. Nach jeder Verschlüsselung wird die Sequenznummer um einen konstanten Wert erhöht, der meist eins beträgt. Der resultierende Chiffreblock wird anschließend mit den Klartextdaten XOR verknüpft. Somit entsteht aus dem Blockalgorithmus ein Stromalgorithmus, der den Vorteil hat, dass kein Sonderfall bei der Verschlüsselung des letzten Chiffreblocks auftritt. Ist der letzte Klartextblock kürzer, so werden nur *n*-Bits des letzten Chiffreblocks zur Verknüpfung genutzt. Sowohl Sender als auch Empfänger müssen allerdings zur Nutzung des Verfahrens den Anfangswert des Zählers und die Inkrementierungsregeln kennen.

Darüber hinaus ist das Verfahren parallelisierbar, da die einzelnen Sequenznummern simultan genutzt, verschlüsselt bzw. entschlüsselt werden können. Dies kommt der heutigen technologischen Entwicklung der Computerhardware entgegen, so dass es besonders effizient umgesetzt werden kann. Letztendlich hat sich dieses Verfahren seit mehr als 25 Jahren bewährt, so dass es unter Kryptographen als vertrauenswürdig angesehen wird.

Offset Codebook Mode

Der Offset Codebook Mode *(OCB)* ist ein von Phillip Rogaway [RBBK01] an der Universität von Kalifornien entwickelter kryptographischer Modus. Im Gegensatz zu den bisher vorgestellten kryptographischen Modi, stellt OCB nicht nur die Vertraulichkeit, sondern gleichzeitig auch die Integrität der Nachricht sicher. Dies macht ihn einfach in der Handhabbarkeit praktischer Umsetzungen, da Vertraulichkeit und Integrität nicht durch separate Mechanismen realisiert werden müssen. Weiterhin ist OCB, genau wie CTR, parallelisierbar und darüber hinaus höchst effizient. Nachfolgend wird die Funktionsweise von OCB erläutert, ohne auf die genauen Abläufe im Einzelnen einzugehen. Weiterführende Informationen, speziell für die Implementierung, können [KR05] entnommen werden.

Zunächst wird der IV mit dem Blockalgorithmus verschlüsselt, sowie die gesamte Nachricht in einzelne Klartextblöcke zerlegt. Dabei erfährt der letzte Klartextblock eine Sonderbehandlung, da er in den überwiegenden Fällen nicht komplett gefüllt ist. Anschließend wird jeder Klartextblock, mit Ausnahme des letzten, XOR mit einer festgelegten Bitsequenz *(Offset)* aus dem zu Beginn verschlüsselten IV einmal vor und einmal nach der Verschlüsselung, mit dem Blockalgorithmus, verknüpft. Gleichzeitig werden die jeweiligen Klartextblöcke XOR verknüpft, um so eine Checksumme für die Integritätssicherung über alle Klartextblöcke zu ermitteln. Ist dies erledigt wird der letzte Klartextblock verschlüsselt. Hierzu wird die binäre Repräsentation der Länge, des im letzten Klartextblock enthaltenden Nachrichtenteils mittels Padding aufgefüllt, mit dem Offset XOR verknüpft und anschließend mit dem Blockalgorithmus verschlüsselt. Das entstandene Resultat entspricht einem Zufallsstrom, analog eines Stromchiffrierungsalgorithmus. Abschließend wird der Zufallsstrom dazu verwendet,

den im letzten Klartextblock enthaltenden Nachrichtenteil zu verschlüsseln, so dass die gesamte Nachricht nach dem zusammenfügen der Chiffreblöcke verschlüsselt ist. Um die Checksumme der Nachricht zu vervollständigen, wird der im letzten Klartextblock enthaltende Nachrichtenteil, mit dem für die Verschlüsselung ungenutzten Teil des Zufallsstroms aufgefüllt und XOR mit der bereits über die vorherigen Klartextblöcke berechneten Checksumme verknüpft. Die so entstandene Checksumme wird mit dem Offset, XOR verknüpft und mit dem Blockalgorithmus ebenfalls verschlüsselt.

Darüber hinaus unterstützt OCB die Möglichkeit einen sogenannten *Header* der Nachricht hinzuzufügen. Als Header wird eine Nachricht bezeichnet, deren Integrität lediglich gesichert werden soll. Wird von dieser Option gebrauch gemacht, wird mittels dem PMAC[1] Verfahren [Rog01] eine Checksumme über den Header bestimmt und mit der Checksumme der eigentlichen Nachricht XOR verknüpft.

Allerdings ist OCB durch Patente geschützt, weswegen es bisher nur eine geringe Verbreitung gefunden hat. So wurde auch das ursprünglich im IEEE 802.11i Standard vorgesehene WRAP[2] Verfahren [NWG02], dass auf dem OCB Modus aufbaut nicht als Verbindlich in den Standard aufgenommen.

2.2.3 Asymmetrische Kryptographie

Die asymmetrische Kryptographie, auch als Public-Key-Kryptographie bekannt, ist ein weiteres Verfahren der Kryptographie. Im Wesentlichen wurden die Grundlagen hierfür von Whitfield Diffie und Martin Hellman [DiHe76] in den 1970er Jahre gelegt.

Asymmetrische Kryptographie arbeitet mit zwei unterschiedlichen Schlüsseln, einer von ihnen ist öffentlich bekannt, der zweite geheim. Der öffentlich bekannte Schlüssel fungiert als Chiffrierschlüssel, während der geheime Schlüssel, auch als privater Schlüssel bezeichnet, als Dechiffrierschlüssel genutzt wird. So können beliebige Parteien Nachrichten verschlüsseln, aber nur die Partei mit Kenntnis des privaten Schlüssels kann die Nachrichten entschlüsseln. Während bei der symmetrischen Kryptographie der Dechiffrierschlüssel aus dem Chiffrierschlüssel und umgekehrt berechnet werden kann, kann bei der asymmetrischen Kryptographie der Dechiffrierschlüssel nicht bzw. nicht in angemessener Zeit aus dem Chiffrierschlüssel berechnet werden. Zur Signierung von digitalen Dokumenten kann bei einigen Algorithmen, z.B. RSA, der private Schlüssel zur Verschlüsselung und öffentliche Schlüssel zur Entschlüsselung genutzt werden. In diesem Fall wird dann von einer *digitalen Signatur* gesprochen.

Ein großer Vorteil der asymmetrischen Kryptographie ist, dass das Schlüsselverteilungsproblem entfällt. So muss kein Schlüsselaustausch über einen sicheren Kanal vorgenommen werden, bevor verschlüsselte Nachrichten ausgetauscht

[1] Parallelizable Message Authentication Code
[2] Wireless Robust Authentication Protocol

werden können. Diesem Vorteil steht allerdings auch ein großer Nachteil gegenüber, so ist der Berechnungsaufwand der benötigt wird, um etwa den Faktor 100 bis 1000 höher.

Um Nachrichten, beispielsweise zwischen den zwei Parteien Alice und Bob, zu übertragen müssen Alice und Bob sich zunächst darauf verständigen asymmetrische Kryptographie zu nutzen. Danach sendet Bob, Alice seinen öffentlichen Schlüssel und Alice, Bob den ihrigen. Nun kann Alice Nachrichten an Bob mit dessen öffentlichen Schlüssel verschlüsseln und so die Nachricht, während der Übertragung, vor unbefugten Zugriff Dritter schützen. Bob seinerseits würde entsprechend den öffentlichen Schlüssel von Alice nutzen. Nach der Übermittlung kann Bob seine Nachricht mit seinem privaten Schlüssel entschlüsseln und den Inhalt lesen. Analog kann Alice ihre Nachricht entschlüsseln und den Inhalt lesen.

2.2.4 Hybride Kryptographie

Bei dieser Art der Kryptographie handelt es sich um eine Verschmelzung zwischen der symmetrischen Kryptographie und der asymmetrischen Kryptographie. Das Ziel ist es die Vorteile beider miteinander so zu verbinden, dass die jeweiligen Schwächen beseitigt werden. So ist der wesentliche Nachteil der symmetrischen Kryptographie, der Schlüsselaustausch vor der Verschlüsselung. Dies ist als Schlüsselverteilungsproblem bekannt. Die asymmetrische Kryptographie hat dieses Problem nicht, benötigt allerdings einen hohen Berechnungsaufwand für die Verschlüsselung und Entschlüsselung der Nachrichten.

Daher nutzt die hybride Kryptographie zunächst asymmetrische Kryptographie, um einen sicheren Kanal aufzubauen. Über den sicheren Kanal werden anschließend die Schlüssel für die symmetrische Kryptographie ausgetauscht, so dass das Schlüsselverteilungsproblem gelöst ist. Anschließend wird symmetrische Kryptographie für die Verschlüsselung und Entschlüsselung der Nachrichten genutzt. Somit wird die Effizienz der symmetrischen Kryptographie, für die Verschlüsselung und Entschlüsselung der Nachrichten verwendet, während das Schlüsselverteilungsproblem mittels asymmetrischer Kryptographie gelöst wird. Zudem wird die Schlüsselverwaltung erleichtert, da die, für die Kommunikation benötigten Schlüssel *(Sitzungsschlüssel)* nicht fest gespeichert werden müssen, sondern bei Bedarf erzeugt werden können. Nachfolgend wird ein Beispiel aufgezeigt, wie eine Kommunikation zwischen den beiden Parteien Alice und Bob mit Hilfe hybrider Kryptographie ablaufen könnte.

Zunächst müssen Alice und Bob sich darauf verständigen hybride Kryptographie nutzen zu wollen und damit einher, eventuell den verwendeten Verschlüsselungsalgorithmus. Bob würde Alice anschließend seinen öffentlichen Schlüssel senden, so dass Alice diesen für die Verschlüsselung nutzen kann. Um eine Manipulation des öffentlichen Schlüssels zu vermeiden, wird der öffentliche Schlüssel in der Praxis häufig digital signiert in eine Datenbank abgelegt. Alice kann nun den symmetrischen Schlüssel für

den eigentlichen Nachrichtenaustausch und falls noch nicht geschehen eine Kennung für den verwendeten Verschlüsselungsalgorithmus verschlüsseln. Dabei muss Alice den Schlüssel nicht gespeichert haben, sondern kann ihn auch zufällig für jede Sitzung neu bestimmen, weshalb dieser Schlüssel als Sitzungsschlüssel bezeichnet wird. Nachdem Alice, Bob den Sitzungsschlüssel gesendet hat, kann dieser seinerseits den Sitzungsschlüssel mit seinem privaten Schlüssel entschlüsseln. Damit ist das Schlüsselverteilungsproblem gelöst und beide können ihre Nachrichten vor unbefugten Dritten schützen, indem sie den vereinbarten Sitzungsschlüssel zur Verschlüsselung bzw. Entschlüsselung der Nachrichten unter Verwendung des vereinbarten symmetrischen Verschlüsselungsalgorithmus nutzen.

2.3 Security in WLANs

2.3.1 Aspekte der Security

Die in *Abschnitt 2.2.1* vorgestellten Aspekte der Security werden nachfolgend im Kontext der IEEE 802.1# Standards näher beleuchtet. Dabei zeigt sich, dass die Aspekte der Security nicht in jedem technischen System sinnvoll umgesetzt werden können.

Authentizität

In einem WLAN-Netzwerk kommunizieren alle Stationen über ein gemeinsames Medium. Die Kommunikationspartner sind daher selbst verantwortlich ihre Authentizität, gegenseitig mittels einer Authentifizierung, zu überprüfen. Dies kann auf unterschiedliche Arten geschehen, beispielsweise über Zertifikate oder über einen gemeinsames Geheimnis, wie es der IEEE 802.11 Standard und der IEEE 802.11i Standard vorsehen. Um eine robuste Authentifizierungsmethode zu erhalten, sollten die folgenden vier Basisanforderungen [EdAr04] erfüllt sein.

1. Keine Möglichkeit die Authentifizierungsmethode zu umgehen
2. Bewahrung und nicht Übertragbarkeit der Identität über eine gewisse Zeit
3. Gegenseitige Authentifizierung
4. Verwendung eines von der Verschlüsselung verschiedenen Schlüssels

Integrität

Eng mit dem Aspekt der Authentizität, ist der Aspekt der Integrität verbunden. So ist es aufgrund des gemeinsam genutzten Mediums nicht ausreichend, lediglich die Identität des Urhebers nachzuweisen. Daher sollte, im Sinne der Security, in einem WLAN-Netzwerk auch die Integritätssicherung der Daten, während der Kommunikation sicher gestellt werden. Sowohl der IEEE 802.11 Standard als auch der IEEE 802.11i Standard legen zur Integritätssicherung der Daten Mechanismen fest. Diese werden in *Abschnitt 2.3.2* näher betrachtet.

Vertraulichkeit

Da während einer Kommunikation Daten mit sensiblen Informationen ausgetauscht werden könnten, z.B. Passwörter für Webseiten, ist dieser Aspekt von besonderem Interesse. So sollen während einer Kommunikation sensible Informationen nicht von unberechtigten Dritten in Erfahrung gebracht werden können.

In einem WLAN-Netzwerk ist die Existenz der Daten, aufgrund des gemeinsam genutzten Mediums zur Kommunikation, nicht vertraulich. Daher werden zur Wahrung der Vertraulichkeit der Informationen, kryptographische Verfahren verwendet. Der IEEE 802.11 Standard und der IEEE 802.11i Standard definieren hierfür Mechanismen, die im *Abschnitt 2.3.2* näher betrachtet werden.

Verfügbarkeit

Der Aspekt der Verfügbarkeit lässt sich in WLAN-Netzwerken nicht gewährleisten. Ursache hierfür ist das gemeinsam genutzte Medium, auf das alle Stationen gleichermaßen Zugriff haben. Damit kann eine nicht autorisierte Station die Kommunikation zwischen anderen Stationen stören, unabhängig ob diese authentifiziert sind oder nicht.

Verbindlichkeit

Bei der Verbindlichkeit muss die Authentizität der Absenderstation und die Integrität der Daten von der Empfängerstation überprüft werden können. Weiterhin muss dies auch zu einem späteren Zeitpunkt, durch einen berechtigten Dritten möglich sein. Dies lässt sich grundsätzlich in WLAN-Netzwerken realisieren, allerdings definiert der IEEE 802.11i Standard keine Mechanismen hierfür. Aus Gründen der Performance wird lediglich ein Mechanismus gegen Replay-Angriffe *(siehe Abschnitt 2.3.4)* und ein Mechanismus zur Überprüfung der Absenderauthentizität für Unicast-Verbindungen definiert, was dem Aspekt der Verbindlichkeit nicht genügt. Soll die Verbindlichkeit dennoch gewährleistet werden, so können zusätzliche Mechanismen oberhalb der MAC-Schicht implementiert werden. In [Tsa04] sind Beispiele für mögliche Implementierungen der Verbindlichkeit auf der Applikationsebene aufgezeigt.

2.3.2 IEEE und Wi-Fi Alliance

Bei der IEEE und der Wi-Fi Alliance handelt es sich um zwei Institutionen, mit verschiedenen Zielsetzungen. Während die historisch gesehen ältere, bereits 1963 gegründete, IEEE sich unter anderem verantwortlich für die Normung von Hardware und Software, Herausgabe wissenschaftlicher Beiträge, sowie Veranstalter diverser Fachtagungen im Bereich der Informatik und Elektrotechnik zeigt, ist die Wi-Fi Alliance eine Vereinigung zahlreicher Hersteller von drahtlosen Geräten, die eine

Zertifizierung von Produkten, mit dem Ziel der Interoperabilität, auf Basis des IEEE 802.11 Standards vornimmt. Die Wi-Fi Alliance wurde 1999 gegründet und ist unter anderem für die Etablierung des Security-Verfahrens Wi-Fi Protected Access *(WPA)* verantwortlich. Nachfolgend werden, neben dem von der Wi-Fi Alliance geprägten WPA Verfahren auch Verfahren der IEEE und ein nicht Standardisiertes Verfahren im Überblick vorgestellt.

WEP

Das WEP[1]-Verfahren war in den ersten fünf Jahren nach der Verabschiedung des IEEE 802.11 Standards die einzige Möglichkeit Nachrichten vor dem Zugriff unberechtigter Dritter zu schützen. Es lässt sich effizient sowohl in Hardware als auch in Software, sowie im Infrastruktur-Modus und Ad-hoc-Modus umsetzen. Das Ziel bei der Konzeption des Verfahrens war es nicht die Security eines Militärsystems bieten zu können, dennoch sollte es schwierig sein aus ein mit WEP gesichertes WLAN-Netzwerk vertrauliche Informationen zu gewinnen oder Manipulationen darin vorzunehmen. Inzwischen gibt es allerdings diverse Werkzeuge mit denen sich die Schwachstellen von WEP *(siehe Abschnitt 2.3.4)* aktiv ausnutzen lassen. Somit kann WEP die Vertraulichkeit und Integrität der Nachrichten nicht länger gewährleisten und sollte daher nicht weiter verwendet werden. Da es jedoch für Ad-hoc-Netzwerke und damit auch für WMNs verfügbar ist, sowie juristisch von Belang ist soll es nachfolgend vorgestellt werden.

Grundsätzlich bedient sich das WEP-Verfahren an zwei Security-Mechanismen [EdAr04], wovon ein Security-Mechanismus die Authentizität gewährleistet und der zweite die Vertraulichkeit, sowie die Integrität. Der Security-Mechanismus zur Authentifizierung *(Challenge-Response-Handshake)* ist dabei nur für den Infrastruktur-Modus konzipiert und zudem optional. Die Authentifizierung erfolgt auf Basis eines geheimen Schlüssels, den sowohl der AP als auch die Station kennen müssen. Zur Wahrung der Vertraulichkeit, sowie der Integrität der Nachricht wird dieser Schlüssel gleichermaßen verwendet. Der Schlüssel hat eine Länge von 40 oder 104 Bit [IEE04, IEE07] und wird vom Anwender konfiguriert. Zur Authentifizierung sendet die Station zunächst eine Anfrage an den AP, der daraufhin eine unverschlüsselte *Challenge*-Nachricht, idealer Weise zufällig bestimmt, an die Station zurücksendet. Die *Challange*-Nachricht, mit einer Länge von 128 Bit, wird nach dem Empfang von der Station mit ihrem geheimen Schlüssel verschlüsselt und an den AP als *Response*-Nachricht zurückgesendet. Dieser entschlüsselt die *Response*-Nachricht mit seinem geheimen Schlüssel und vergleicht das Resultat mit der zuvor gesendeten *Challenge*-Nachricht. Stimmen beide überein ist die Station beim AP authentifiziert und der AP sendet eine Bestätigung.

Die Verwendung eines gemeinsamen Schlüssels zur Authentifizierung, sowie zur Wahrung der Vertraulichkeit und Integrität macht den *Challenge-Response-Handshake*

[1] Wired Equivalent Privacy

optional. Ist der geheime Schlüssel zur Entschlüsselung verschieden dem zur Verschlüsselung, so tritt unabhängig davon ob eine Authentifizierung durchgeführt wurde oder nicht ein Fehler auf und die Nachricht wird verworfen. Wird der Challenge-Response-Handshake nicht durchgeführt, wird eine sogenannte *Offene Authentifizierung* durchgeführt. Hierbei ist der Begriff: *„Offene Authentifizierung“* irreführend, denn es wird keine Authentifizierung im eigentlichen Sinne vorgenommen. Vielmehr handelt es sich um eine Art Bekanntmachung, indem die Station sich durch senden einer Anfrage an den AP bekannt macht. Dieser bestätigt abschließend den Erhalt der Anfrage, womit die *Offene Authentifizierung* vollzogen ist.

Durch die Verwendung des von Ron Rivest 1987 entwickelten RC4-Algorithmus, einem Stromchiffrierungsalgorithmus, zur Verschlüsselung der Nachrichten wird die Vertraulichkeit und Integrität sichergestellt. Dazu wird zunächst ein Integrity Check Value *(ICV)*, zur Gewährleistung der Integrität, mittels einer CRC[1]32 Checksumme [Sch00] über die gesamte Nachricht bestimmt. Anschließend wird dieser an die Nachricht angehängt und gemeinsam mit der Nachricht, unter Nutzung des Schlüssels, der sich aus einem 24 Bit langen IV, sowie dem geheimen Schlüssel zusammensetzt und dem RC4-Algorithmus verschlüsselt. Ist dies geschehen, so kann die Nachricht mit dem IV vorangestellt versendet werden. Weiterhin bietet WEP die Möglichkeit bis zu vier Schlüssel manuell festzulegen und über Indices zu verwalten. Damit soll dem erhöhten Security-Risiko bei häufiger Verwendung nur eines Schlüssels begegnet werden. Diese Problematik wird später, in *Abschnitt 2.3.3; Schlüsselmanagement*, genauer beleuchtet.

IEEE 802.11i

Nachdem abzusehen war, dass das WEP-Verfahren den Security-Anforderungen nicht genügen kann, wurde die Arbeit an dem IEEE 802.11i Standard aufgenommen. Er wurde 2004 verabschiedet, ist seit 2007 fester Bestandteil des IEEE 802.11 Standards [IEE07] und bietet bedeutende Verbesserungen in der Security von WLAN-Netzwerken. Die neuen Security-Mechanismen werden in der Kategorie Robust Security Network Association *(RSNA)* zusammengefasst, während nahezu alle Security-Mechanismen des WEP-Verfahrens zur Kategorie Pre-RSNA gehören und damit nur aus Gründen der Abwärtskompatibilität weiterhin aufgeführt werden. Einzig der keine Security gewährleistende Mechanismus, der Offenen Authentifizierung wird der Kategorie RSNA zugeteilt.

Die RSNA-Mechanismen umfassen neue Authentifizierungsverfahren auf Basis der portbasierten Zugangskontrolle des IEEE 802.1X Standards [IEE04X], ein Schlüsselmanagement, neue Verschlüsselungsprotokolle, sowie ein neues Verschlüsselungsverfahren. Zu den Authentifizierungsverfahren gehören eine Authentifizierung mittels Authentifizierungsserver, z.B. einem RADIUS[2]-Server und

[1] Cyclic Redundancy Check
[2] Remote Authentication Dial-In User Service

eine Authentifizierung mittels Pre-Shared Key *(PSK)* unter Nutzung des 4-Wege-Handshake Protokolls. Auf beide Verfahren, das Schlüsselmanagement, sowie die Verschlüsselungsprotokolle Temporal Key Integrity Protocol *(TKIP)* und Counter Mode-CBC Message Authentication Code Protocol *(CCMP)*, mit dem neu aufgenommenen Verschlüsselungsverfahren Advanced Encryption Standard *(AES)* [NIST01], wird in *Abschnitt 2.3.3* näher eingegangen. Auch existieren auf Basis des IEEE 802.11i Standards weitere Security-Konzepte, wie beispielsweise in [RWSX07] beschrieben, die spezielle Anwendungsbezogene Security-Anforderungen erfüllen.

WPA

Als erfolgreiche Angriffe auf die Schwachstellen des WEP-Verfahrens *(siehe Abschnitt 2.3.4)* bekannt wurden, wurde noch aktiv am IEEE 802.11i Standard gearbeitet. Eine Verabschiedung war zu diesem Zeitpunkt noch nicht absehbar, so dass die Wi-Fi Alliance nach einer raschen Lösung strebte. Das Ergebnis dieser Bemühungen ist WPA [WiFi03].

Die WPA-Zertifizierung basiert auf den Entwurf IEEE 802.11i / D3.0 und ist damit ein Vorgriff auf den IEEE 802.11i Standard. WPA ist darauf ausgelegt auf der bereits von WEP genutzten Hardware realisiert zu werden. Weiterhin sind die verwendeten Security-Mechanismen ebenfalls Teil des IEEE 802.11i Standards, so dass es einen Zwischenschritt von WEP, hin zum vollständigen IEEE 802.11i Standard bildet. Zur Beseitigung der Schwächen von WEP beinhaltet WPA das verbesserte Verschlüsselungsprotokoll TKIP, sowie neue Authentifizierungsverfahren. Diese werden in *Abschnitt 2.3.3*, zusammen mit bedeutenden Mechanismen des IEEE 802.11i Standards, vorgestellt.

WPA2

Nachdem der IEEE 802.11i Standard im Juni 2004 vollständig vorlag, führte die Wi-Fi Alliance im September 2004 eine neue Zertifizierung, mit dem Namen Wi-Fi Protected Access Version 2 *(WPA2)* [WiFi05], ein. Sie beinhaltet alle vorgeschriebenen Mechanismen des IEEE 802.11i Standards und wird daher oft als Synonym für selbigen verwendet.

Um die Umstellung von WPA auf WPA2 zu erleichtern ist WPA2 abwärtskompatibel zu WPA. So bietet WPA2 einen Mixed Mode an, der das gleichzeitige Betreiben von Geräten mit WPA- und WPA2-Unterstützung ermöglicht.

WPA-None

Das WPA-None-Verfahren ist in keinem Standard festgelegt. Vielmehr stellt es, im Vergleich zu WEP, einen für Ad-hoc-Netzwerke verbesserten Mechanismus dar. So wird der geheime Schlüssel nicht direkt genutzt, sondern eine statische Ableitung zur

Verschlüsselung bestimmt. Für die Datenverschlüsselung selbst kommen die Verschlüsselungsprotokolle TKIP oder optional CCMP des IEEE 802.11i Standards zum Einsatz, so dass die Daten durch ein stärkeres Kryptographieverfahren geschützt sind. Auf einen Mechanismus zur Authentifizierung wird hingegen verzichtet, so dass es unter Berücksichtigung der einmaligen statischen Ableitung des Schlüssels, verglichen mit den im IEEE 802.11i Standard zur Verfügung stehenden Security-Mechanismen, nur ungenügende Security bieten kann.

2.3.3 Security-Mechanismen des IEEE 802.11i Standards

Der vorangegangene *Abschnitt 2.3.2* stellte diverse Security-Verfahren der IEEE und der Wi-Fi Alliance, sowie ein nicht standardisiertes Security-Verfahren vor. Diese Security-Verfahren bauen im Wesentlichen auf gleiche Security-Mechanismen auf. Nachfolgend wird auf diese Security-Mechanismen im Einzelnen eingegangen.

TKIP

Das Temporal Key Integrity Protocol *(TKIP)* ist Bestandteil des IEEE 802.11i Standards, sowie Bestandteil der WPA- und WPA2-Zertifizierung der Wi-Fi Alliance. Es handelt sich um ein Verschlüsselungsprotokoll, welches geringe Anforderungen an die Ressourcen stellt und dennoch ein hohes Maß an Security bietet. Durch die geringe Anforderung an die Ressourcen kann es mit der zu WEP kompatiblen Hardware genutzt werden.

Um eine Wiederverwendung und damit die Erzeugung eines gleichen Schlüssels als mögliche Schwäche zu verhindern, beträgt die Größe des IVs nunmehr 48 Bit. Der IV beginnt bei Null und wird für jedes Datenframe um eins inkrementiert. Dies erlaubt gleichzeitig die Verwendung des IVs als TKIP Sequence Counter *(TSC)*, so dass damit ein Mechanismus gegen Replay-Angriffe *(siehe Abschnitt 2.3.4)* zur Verfügung steht. Es werden hierzu alle Frames, deren IV kleiner oder gleich dem aktuellen IV ist verworfen.

Zur Sicherung der Integrität der Nachrichten wird das effizient im Treiber implementierbare Nachrichtenintegritätsverfahren Michael verwendet. Es berechnet, im Gegensatz zu vielen anderen Integritätsverfahren, den MIC für die gesamte Nachricht, statt für jedes Frame separat. Allerdings ist Michael aufgrund seiner Einfachheit anfällig für Brute-Force-Angriffe, so dass es zusätzlich einen Mechanismus mit dem Namen *Countermeasures*, zum erkennen eines Angreifers und zum einleiten entsprechender Gegenmaßnahmen, umsetzt. Um der Vertraulichkeit der Nachrichten zu gewährleisten, wird für jedes Frame ein neuer Schlüssel aus dem Sitzungsschlüssel *(siehe Schlüsselmanagement)* erzeugt. Diese dynamischen Schlüssel bringen dem verwendeten RC4-Stromchiffrierungsverfahren und damit dem Verfahren zusätzliche Stärke ein.

CCMP

Das Counter Mode-CBC Message Authentication Code Protocol *(CCMP)* ist fester Bestandteil des IEEE 802.11i Standards und der WPA2-Zertifizierung, der Wi-Fi Alliance. Es ist mit dem Ziel entwickelt worden keine Kompromisse, hinsichtlich der Security in der Verschlüsselungsphase, einzugehen und damit als Standardprotokoll des IEEE 802.11i Standards Verwendung zu finden.

CCMP nutzt den von Joan Daemen und Vincent Rijmen [DaRi99] entwickelten Blockchiffrierungsalgorithmus AES, um die Vertraulichkeit, sowie die Integrität der Nachrichten zu gewährleisten. Die Schlüssel- und Blocklänge ist dabei auf 128 Bit festgelegt. Um die Integrität zu gewährleisten, wird ein MIC mit Hilfe des CBC Message Authentication Code über das Frame bestimmt.

Die Sicherung der Vertraulichkeit eines Frames erfolgt hingegen über eine anderen kryptographischen Modus, den CTR Mode. Im CTR Mode wird der IV, mit einer Länge von 48 Bit, geschützt durch den Sitzungsschlüssel *(siehe Schlüsselmanagement)* als *Nonce-Wert* verwendet. Als Nonce-Wert wird dabei ein Zahlenwert bezeichnet, der mit einem Schlüssel nur selten, idealer Weise nur einmal verwendet werden darf. Somit existiert gleichermaßen ein Mechanismus zum Schutz vor Replay-Angriffen *(siehe Abschnitt 2.3.4).*

Schlüsselmanagement

Das WEP-Verfahren gibt Anwendern bereits die Möglichkeit manuell bis zu vier Schlüssel zu konfigurieren. So ist der Anwender dafür verantwortlich von Zeit zu Zeit die verwendeten Schlüssel zu wechseln, um dem Problem der Schlüsselalterung zu entgehen. Die Verwendung eines Schlüssels über einen längeren Zeitraum *(Schlüsselalterung)* bedeutet, dass die Wahrscheinlichkeit einer Kompromittierung steigt, da Schlüssel beispielsweise notiert werden oder verloren gehen. Darüber hinaus lohnt es sich für einen Angreifer eher, auch aufwendigere Angriffe, z.B. Brute-Force-Angriffe, zur Kompromittierung des Schlüssels durchzuführen. So können mehr Informationen gewonnen werden, was auf Seiten des Anwenders gleichzeitig einen höheren Schaden verursacht. Auch wird damit die Kryptoanalyse erleichtert, weil eine Analyse von mehr Chiffretext, verschlüsselt mit dem gleichen Schlüssel, generell einfacher ist [Sch96].

Allerdings zeigt sich, dass die Möglichkeit manuell Schlüssel zu wechseln von Anwendern nur selten genutzt wird. Dafür gibt es seitens der Anwender zahlreiche Gründe, so ist beispielsweise der administrative Aufwand besonders in großen WLAN-Netzwerken hierfür recht hoch oder es wird schlicht und einfach vergessen. Zwar stehen auch Lösungen zur Schlüsselverteilung, z.B. Cisco LEAP [Kin01], zur Verfügung allerdings ist auch hier der Aufwand der Konfiguration und Aufrechterhaltung recht hoch. Dies wird im IEEE 802.11i Standard, sowie in der WPA-Zertifizierung der Wi-Fi Alliance zum einen durch die Einführung eines Authentifizierungsserver *(siehe Upper-Layer Authentifizierung)* und zum anderen durch Verwendung eines PSK für den

Anwender vereinfacht. Um das Eingangs erwähnte Problem der Schlüsselalterung zu beseitigen, sieht der IEEE 802.11i Standard und die WPA-Zertifizierung im Rahmen eines Schlüsselmanagements die Erzeugung einer Schlüsselhierarchie vor. Diese beinhaltet eine explizite Unterscheidung von Unicast- und Multicast-Verbindungen. So entsteht eine Unicast-Verbindung, wenn genau zwei Kommunikationspartner direkt kommunizieren. In einem solchen Fall sollte die Authentizität, die Vertraulichkeit und Integrität der Nachrichten für diese Verbindung gewahrt werden. Auf Multicast- und Broadcast-Verbindungen bezogen bedeutet dies, dass nur eine gewisse Anzahl vertrauenswürdiger Stationen, die Nachrichten lesen können, sollen. Zu diesem Zweck führt der IEEE 802.11i Standard, sowie WPA eine Unterscheidung von Schlüsseln ein. Schlüssel die zur Sicherung von Unicast-Verbindungen verwendet werden, werden als paarweise Schlüssel *(PMK*[1]*)* bezeichnet und Schlüssel zur Sicherung von Multicast-Verbindungen als Gruppenschlüssel *(GMK*[2]*).* Um der Problem der Schlüsselalterung zu begegnen, werden weitere Schlüssel daraus abgeleitet, die dann von Zeit zu Zeit erneuert werden können *(Rekeying).* Die abgeleiteten Schlüssel sollen dabei möglichst keinen offensichtlichen Bezug zueinander haben. So können für die verschiedenen Security-Mechanismen, verschiedene Schlüssel verwendet werden. Dies stellt einen deutlichen Gewinn an Security dar. Nachfolgend wird zunächst die Schlüsselhierarchie für die paarweise Kommunikation genauer betrachtet, bevor auf die Schlüsselhierarchie für Multicast-Verbindungen eingegangen wird. Abbildung 2.1 veranschaulicht die Schlüsselhierarchie für die paarweise Kommunikation nach dem IEEE 802.11i Standard.

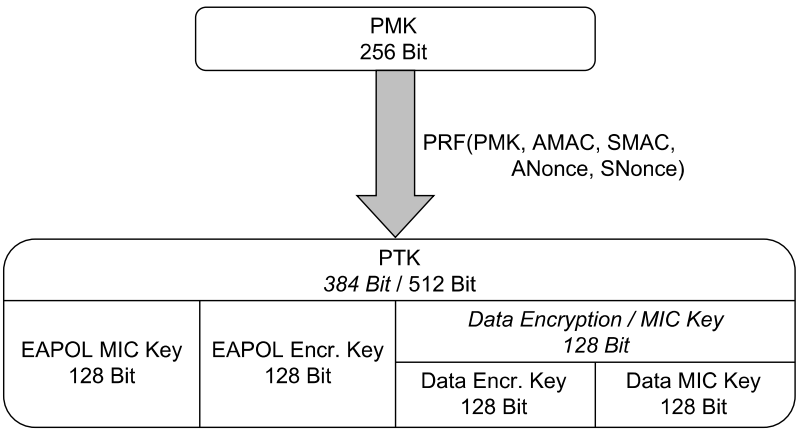

Abbildung 2.1: Schlüsselhierarchie für Paarweise Schlüssel

Der PMK dient einerseits als Ausgangspunkt zur Ableitung weiterer Schlüssel, für die paarweise Kommunikation, und anderseits zur Authentifizierung. Er wird abhängig von der gewählten Authentifizierungsmethode bestimmt. Wird eine Upper-Layer Authentifizierungsmethode verwendet, so bestimmt diese den PMK. Bei der Nutzung eines PSK zur Authentifizierung konfiguriert der Anwender den PSK, der dann gleichzeitig dem PMK entspricht. Da ein PSK mit einer Länge von 256 Bit schwierig zu

[1] Pairwise Master Key
[2] Group Master Key

merken ist sieht der IEEE 802.11i Standard und WPA, Mechanismen auf Basis von Hash-Funktionen vor, die den PSK zum einen aus einer Texteingabe erzeugen können und zum anderen eine etwaige Auffüllung vornehmen können.

Ist der PMK bekannt kann die Authentifizierung mit Hilfe des 4-Wege-Handshake Protokolls *(siehe 4-Wege-Handshake)* vorgenommen werden. Dieses Protokoll beinhaltet die Ableitung eines Sitzungsschlüssels *(PTK[1])*, unter Nutzung einer Zufallsfunktion *(PRF)*, für die paarweise Kommunikation. Zur besseren Unterscheidung der Parameter der Zufallsfunktion, bekommen Parameter der gleichen Station den gleichen Präfix dessen Bedeutung im Rahmen des 4-Wege-Handshake geklärt wird. Neben dem PMK benötigt die Zufallsfunktion von beiden Stationen die MAC-Adresse *(AMAC bzw. SMAC)*, sowie die bedeutenden Werte *ANonce* und *SNonce*. Die beiden Nonce-Werte haben eine Länge von jeweils 256 Bit und sollen Rückschlüsse aus dem PTK auf den PMK vermeiden. Zur Erzeugung muss eine kryptographisch hochwertige Zufallsfunktion verwendet werden, so dass der IEEE 802.11i Standard entsprechend hierfür eine Funktion anbietet.

Der erzeugte PTK wird bereits im Verlauf der Authentifizierung für die verschiedenen Aufgaben in weitere, jeweils 128 Bit lange, Schlüssel fragmentiert und hat, abhängig vom verwendeten Verschlüsselungsprotokoll, eine Länge von 384 Bit für CCMP bzw. 512 Bit für TKIP. Bei der Fragmentierung des PTKs entsteht so ein EAPOL MIC Key, dessen Aufgabe es ist die Integrität, unter Nutzung des HMAC-MD5 [Riv92, CG97, KBC97] Verfahrens oder des HMAC-SHA1 [CG97, KBC97, NIST02] Verfahrens für das 4-Wege-Handshake Protokoll und das Group Key-Handshake Protokoll *(siehe Group Key-Handshake)*, sicher zu stellen. Soll die Vertraulichkeit von Informationen innerhalb der Handshake-Protokolle gewährleistet werden, so wird hierzu der EAPOL Encryption Key verwendet. Die übrigen 128 bzw. 256 Bit werden schließlich als Sitzungsschlüssel für den Nachrichtenaustausch in CCMP bzw. TKIP verwendet.

Das Konzept der Schlüsselhierarchie für Multicast-Verbindungen hingegen ist unabhängig zur vorangegangenen Betrachtung der Schlüsselhierarchie von Unicast-Verbindungen, nicht aber zur Authentifizierung. So sieht der IEEE 802.11i Standard, wie auch WPA für Multicast-Verbindungen keine erneute Authentifizierung vor. Vielmehr wird eine vorangegangene erfolgreiche paarweise Authentifizierung vorausgesetzt. Mit dem Wegfall der Authentifizierung kann der GMK daher direkt, unter Nutzung einer kryptographisch starken Zufallsfunktion, erzeugt und aktualisiert werden, was den Konfigurationsaufwand für den Anwender reduziert. Er hat analog zum PMK eine Länge von 256 Bit und bildet ebenfalls den Ausgangsschlüssel für die Bestimmung eines weiteren Sitzungsschlüssels *(GTK[2])*. Allerdings wird der Sitzungsschlüssel in diesem Fall für eine Gruppe von Stationen verwendet. Für die vorangehende Erzeugung des GMKs, sowie für die Bestimmung des GTKs, ist der AP verantwortlich. Mit Hilfe des Group Key-Handshake Protokolls *(siehe Group Key-Handshake)*, welches auf eine erfolgreiche paarweise Authentifizierung baut, wird der

[1] Pairwise Transient Key
[2] Group Transient Key

GTK, gesichert durch den EAPOL Encryption Key, an die sich zuvor authentifizierte Station übertragen. Anschließend steht der GTK dem Verschlüsselungsprotokoll CCMP bzw. TKIP zur Sicherung von Multicast-Verbindungen direkt zur Verfügung.

Da mit der Einführung der Schlüsselhierarchien jede Station mindestens zwei Schlüssel *(PTK und GTK)* verwalten muss, ist eine Schlüsselzuordnung zwischen Sender- und Empfängerstation notwendig. Hierzu sieht der IEEE 802.11i Standard und die WPA-Zertifizierung unter anderem die Verwendung von Schlüsselindices, analog zu WEP vor. Die Senderstation trägt den verwendeten Schlüsselindex in den Header des Frames ein, so dass die Empfängerstation über den Schlüsselindex den Schlüssel zum Entschlüsseln der Nachricht kennt. Eine weitere Möglichkeit besteht in der Verwendung eines Schlüsselcaches, indem dem Schlüssel die MAC-Adresse des Absenders zugeordnet wird. Dies wird in der Umsetzung des APs benötigt, da dieser für jede Unicast-Verbindung einen separaten Schlüssel verwalten muss. In Abbildung 2.2 wird exemplarisch der Aufbau eines Frames[1], inklusive der Schlüsselindexierung für das Verschlüsselungsprotokoll CCMP gezeigt.

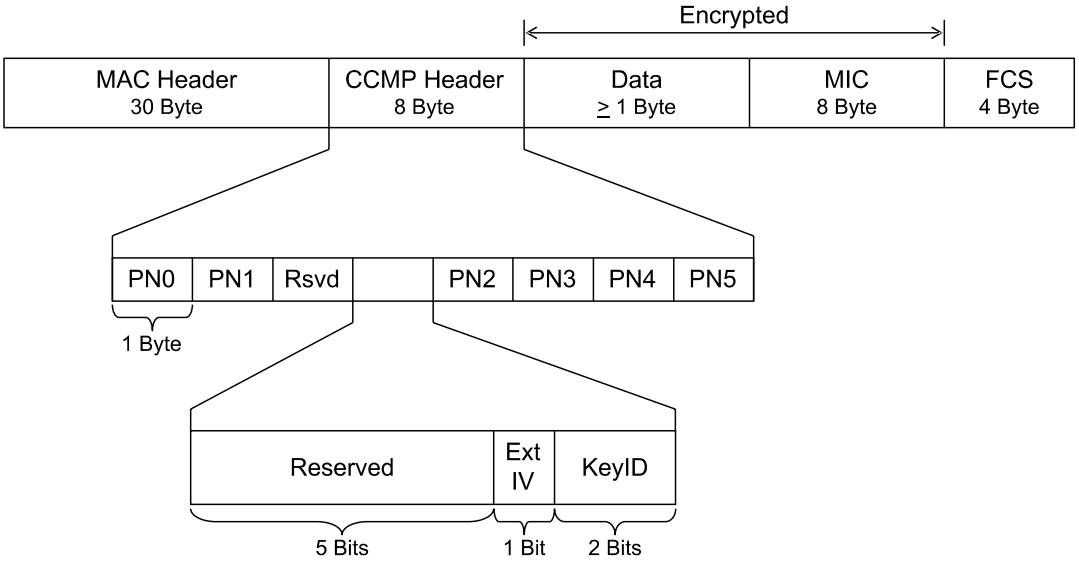

Abbildung 2.2: Aufbau eines Frames unter Nutzung von CCMP[2]

Der Schlüsselindex *(KeyID)* hat, analog zu den anderen Verschlüsselungsprotokollen TKIP und WEP eine Länge von 2 Bit, die neben dem IV *(PN0 bis PN5)* im CCMP Header abgelegt werden. Somit können bis zu vier Schlüssel indiziert werden. Unabhängig vom Verschlüsselungsprotokoll enthält jedes Frame einen Bereich für die zu übertragenden Daten (Data)[3], eine CRC32-Prüfsumme *(FCS)* um Übertragungsfehler festzustellen, sowie einen MAC Header. Der MAC Header enthält hauptsächlich Information für das Routen, z.B. die Empfängeradresse des Frames.

1 Auch als MAC Packet Data Unit *(MPDU)* bezeichnet.
2 Vgl.: [IEE04]
3 CCMP und TKIP nutzen 8 Byte zur Integritätssicherung der Daten *(MIC)*.

4-Wege-Handshake

Wie bereits im vorangegangenen Abschnitt erwähnt ist das 4-Wege-Handshake Protokoll, ein Protokoll zur gegenseitigen Authentifizierung zwischen einer Station und einem AP, auf Basis des PMKs. Gleichzeitig werden die für die Verschlüsselungsprotokolle benötigten Sitzungsschlüssel für die paarweise Kommunikation erzeugt. Es ist im IEEE 802.11i Standard definiert, sowie in der WPA-Zertifizierung enthalten. Zudem stellt es einen bedeutenden Pfeiler des zugrundeliegenden Security-Konzeptes dar.

Zunächst muss im Kontext der Authentifizierung die Aufgabenverteilung zwischen Station und AP klar festgelegt werden. Der AP stellt gewöhnlich anderen Stationen, in Form einer Verbindung zu einem weiteren Netzwerk, einen Dienst zur Verfügung. Um anderen Stationen Zugriff auf diesen Dienst ermöglichen zu können, muss er anderen Stationen aus Sicht der Security eine Authentifizierung ermöglichen. Ein AP oder allgemeiner gefasst eine Station die anderen Stationen eine Authentifizierung ermöglicht, wird fortan als *Authenticator* bezeichnet. Die Station, die diesen Dienst des Authenticators in Anspruch nehmen will, muss sich entsprechend beim Authenticator authentifizieren. Eine solche Station, die sich um eine Authentifizierung durch den Authenticator bemüht, wird fortan als *Supplicant* bezeichnet.

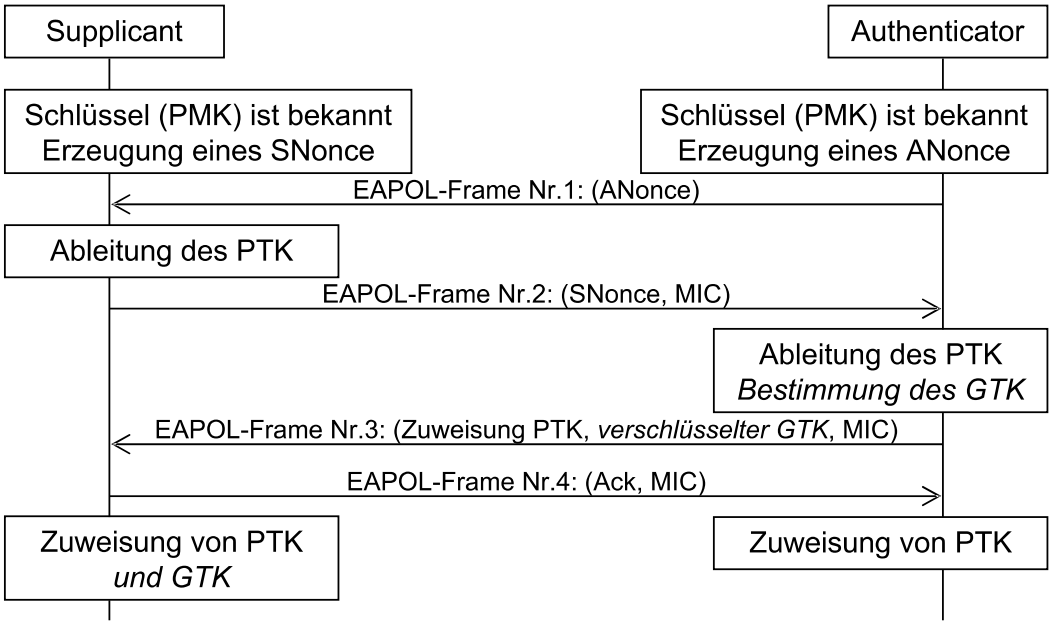

Abbildung 2.3: Das 4-Wege-Handshake Protokoll

Das 4-Wege-Handshake Protokoll sieht einen Nachrichtenaustausch ausgehend vom Authenticator, zur Authentifizierung, vor. Daher geht dem 4-Wege-Handshake eine Offene Authentifizierung zwischen Supplicant *(Station)* und Authenticator *(AP)* gemäß dem IEEE 802.11 Standard voran. Um den Dienst des Authenticators in Anspruch nehmen zu können, muss anschließend eine Zuordnung zwischen Authenticator und Supplicant erfolgen. Dies geschieht mit einer expliziten Anmeldung *(Assoziierungs-Anforderung),* seitens des Supplicants an den Authenticator. Akzeptiert der

Authenticator diese Anforderung, so können beide Parteien miteinander kommunizieren und der Authenticator kann mit der Initiierung des 4-Wege-Handshakes, gemäß Abbildung 2.3, beginnen. Zur Übertragung aller notwendigen Informationen im Verlauf des 4-Wege-Handshakes wird das Extensible Authentication Protocol Over LAN *(EAPOL)*, eine Erweiterung des Extensible Authentication Protocols *(EAP)* [NWG04], genutzt.

Voraussetzung für eine erfolgreiche Authentifizierung zwischen Supplicant und Authenticator ist die Kenntnis des PMKs. Ist der PMK zwischen Authenticator und Supplicant verschieden, so schlägt entsprechend die Authentifizierung fehl. Der Authenticator erzeugt, unter Nutzung einer kryptographisch starken Zufallsfunktion, einen ANonce-Wert, um ihn dem Supplicant mitzuteilen. Dieser wird dem Supplicant im ersten EAPOL-Frame ohne Verwendung von Security-Mechanismen mitgeteilt. Dies ist möglich, da der ANonce-Wert ohne Kenntnis des PMKs nutzlos ist. Darüber hinaus erhält der Supplicant die MAC-Adresse des Authenticators, da diese im MAC-Header des Frames enthalten ist. Damit kann der Supplicant, gemäß der Schlüsselhierarchie für die paarweise Kommunikation, seinen PTK mit Hilfe einer Zufallsfunktion ermitteln. Nun benötigt der Authenticator einen SNonce-Wert des Supplicants, um seinerseits den PTK bestimmen zu können. Dazu sendet der Supplicant seinen, ebenfalls durch eine kryptographisch starke Zufallsfunktion, bereits erzeugten SNonce-Wert im zweiten EAPOL-Frame an den Authenticator. Im Gegensatz zum ersten EAPOL-Frame wird allerdings zur Sicherung der Integrität, ein Message Integrity Code *(MIC)* mittels HMAC-MD5 oder HMAC-SHA1, in Kombination mit einem aus dem PTK abgeleiteten Schlüssel, hinzugefügt. Dies ist notwendig, da im Falle einer Manipulation des SNonce-Wertes, Angriffe auf das Authentifizierungsverfahren möglich wären.

Hat der Authenticator das zweite EAPOL-Frame erhalten, so kann er seinerseits analog zum Supplicant den PTK ermitteln und die Integrität der erhaltenen Informationen überprüfen. Ist die Integrität gewahrt, so sendet der Authenticator in einem dritten EAPOL-Frame dem Supplicant eine Bestätigung. Zusätzlich enthält das EAPOL-Frame für die spätere paarweise Kommunikation in jedem Fall den Initialisierungswert des Sequenzzählers, sowie einen MIC zur Sicherung der Integrität. Abschließend bestätigt der Supplicant den Erhalt des dritten EAPOL-Frames, in einem vierten EAPOL-Frame, wenn die Überprüfung der Integrität zuvor erfolgreich verlief. Damit ist der PTK für die Sicherung der Unicast-Verbindung etabliert und die Voraussetzung für eine Verteilung des GTKs, für Multicast-Verbindungen, mittels des Group Key-Handshakes gegeben.

Group Key-Handshake

Das Group Key-Handshake Protokoll wird zur Verteilung des GTKs verwendet. Es ist im IEEE 802.11i Standard, sowie in der WPA-Zertifizierung festgelegt. Im Gegensatz zum 4-Wege-Handshake Protokoll überprüft es nicht die Authentizität der Kommunikationspartner und setzt daher die erfolgreiche Durchführung des 4-Wege-

Handshake Protokolls voraus. Auf der anderen Seite müssen so weniger Nachrichten ausgetauscht werden, was der Effizienz der Schlüsselaktualisierung zugute kommt.

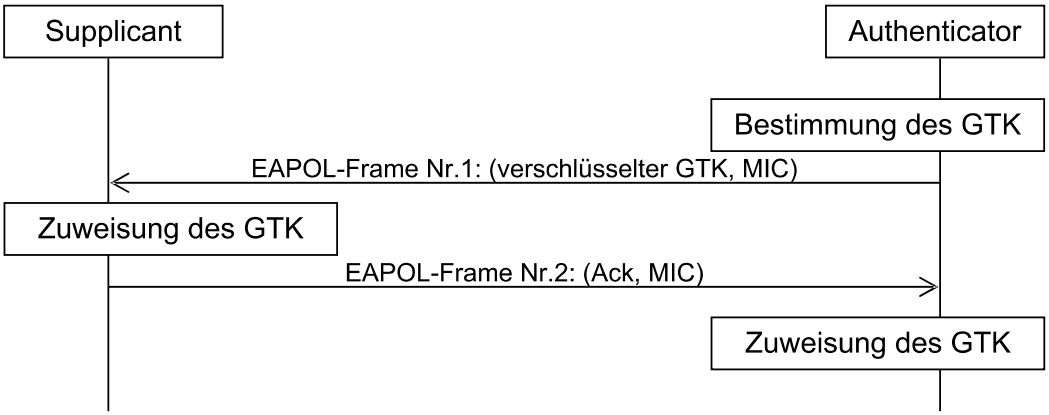

Abbildung 2.4: Das Group Key-Handshake Protokoll

Für die Erzeugung des GTKs ist der Authenticator *(AP)* verantwortlich, so dass er wie in Abbildung 2.4 dargestellt auch den Group Key-Handshake initiiert. Im ersten EAPOL-Frame wird dazu der erzeugte GTK, zur Wahrung der Vertraulichkeit mit dem EAPOL Encryption Key und der Integrität mit dem EAPOL MIC Key, gesichert und an den Supplicant gesendet. Nachdem der Supplicant das EAPOL-Frame erhalten hat, den GTK erfolgreich entschlüsselt und die Integrität überprüft hat, verwendet er ihn fortan zur Entschlüsselung von Multicast-Nachrichten. Abschließend bestätigt der Supplicant in einem zweiten EAPOL-Frame, dass zur Wahrung der Integrität ebenfalls einen MIC enthält, dies dem Authenticator. Mit der Bestätigung ist es dem Authenticator möglich, fortan den GTK zur Verschlüsselung von Multicast-Nachrichten zu verwenden.

Die Durchführung eines Group Key-Handshakes unmittelbar nach der erfolgreichen Durchführung eines 4-Wege-Handshakes kann vermieden werden. So ist eine Verteilung des GTKs bereits innerhalb des dritten EAPOL-Frames, des 4-Wege-Handshakes *(siehe Abbildung 2.3)*, möglich. In diesem Fall wird zur festgelegten Integritätssicherung, analog zum ersten EAPOL-Frame des Group Key-Handshakes, eine Verschlüsselung des GTKs, zur Wahrung der Vertraulichkeit, vorgenommen.

Upper-Layer Authentifizierungsmethoden

Unter der Zusammenfassung Upper-Layer Authentifizierungsmethoden fallen bereits etablierte Authentifizierungsmethoden, oberhalb der Schicht zwei des ISO/OSI-Referenzmodells, die im IEEE 802.11i Standard und der WPA-Zertifizierung vorgesehen sind. Diese sollen es vornehmlich Firmen erleichtern große WLAN Netzwerke zentral zu verwalten. Dabei ist der IEEE 802.11i Standard darauf ausgelegt eine hohe Interoperabilität zu gewährleisten, indem die Möglichkeit eingeräumt wird verschiedene Protokolle, beispielsweise das Transport Layer Security *(TLS)* Protokoll [NWG06] oder das Kerberos Protokoll [NWG05], zusammen mit einem Authentifizierungsserver zu nutzen.

2.3.4 Angriffsmöglichkeiten

Im Jahr 2000, drei Jahre nach Verabschiedung des IEEE 802.11 Standards, wurden die ersten Designschwächen von WEP bekannt. Beim IEEE 802.11i Standard hingegen, konnten in den vergangenen vier Jahren keine Designschwächen ausfindig gemacht werden. Anders als das Verschlüsselungsprotokoll CCMP, ist TKIP mit dem Anspruch entwickelt wurden die Security für ältere, ursprünglich für WEP konzipierte, WLAN-Geräte zu erhöhen, so dass TKIP eine zu WEP vergleichbare Schwäche aufweist. Diese Schwäche kann aber in diesem Kontext nicht als Designschwäche bezeichnet werden, da sie im Rahmen der Zielsetzung von TKIP als Kompromiss anzusehen ist. Neben der Schwäche von TKIP, gibt es einen weiteren bekannten Angriff auf den IEEE 802.11i Standard, unabhängig vom verwendeten Verschlüsselungsprotokoll. Dieser wird neben einigen bekannten Angriffen auf WEP nachfolgend vorgestellt.

WEP

WEP hat in allen für WLAN relevanten Security-Aspekten *(siehe Abschnitt 2.3.1)* Schwächen, welche sich für Angriffe ausnutzen lassen. Um die Authentizität der Stationen zu überprüfen bietet WEP einen Challenge-Response-Handshake an, welcher in *Abschnitt 2.3.2* bereits beschrieben wurde. Damit der Challange-Response-Handshake eine robuste Authentifizierungsmethode für WLAN-Netzwerke darstellt, muss er die vier Basisanforderungen aus *Abschnitt 2.3.1* erfüllen.

Alle vier Anforderungen werden vom Challange-Response-Handshake nicht erfüllt. Ein Angreifer kann so beispielsweise die unverschlüsselte *Challange*-Nachricht mithören und die korrespondierende Response-Nachricht. Durch ein einfaches XOR kann der Angreifer den Schlüsselstrom extrahieren. Nun kann ein Angreifer sich authentifizieren, indem er eine *Challange*-Nachricht anfordert, diese mit dem zuvor extrahierten Schlüsselstrom XOR verknüpft und mit dem IV, der mitgehörten Nachricht, als *Response*-Nachricht an den AP sendet. Somit ist eine Authentifizierung ohne Kenntnis des eigentlichen Schlüssels möglich, womit das Authentifizierungsverfahren seinen Zweck verfehlt.

Zur Sicherung der Integrität von Nachrichten wird einzigst die CRC32-Checksumme verwendet. Diese CRC32-Checksumme wird am Ende der Nachricht angehängt und zusammen mit der Nachricht verschlüsselt. Um eine Nachricht zu manipulieren muss der Informationsteil des Chiffretextes, sowie die CRC32-Checksumme entsprechend angepasst werden. Zur Verschlüsselung wird ein Stromchiffrierungsalgorithmus verwendet, damit hat die Änderung eines Bits an der Position *n* im Klartext, die Änderung des äquivalenten Bits an der Position *n* im Chiffretext zur Folge. In [BGW01] wird unter anderem beschrieben, wie sich aus einer Änderung eines Bits in einer Nachricht, die Änderung in der korrespondierenden CRC32-Checksumme berechnen lässt. Damit kann ein Angreifer, unabhängig ob die Nachricht verschlüsselt ist oder nicht ein Bit im Informationsteil der Nachricht verändern und die CRC32-Checksumme

entsprechend anpassen. Eine Manipulation kann somit nicht zuverlässig erkannt werden, so dass die Sicherung der Integrität nicht gegeben ist.

Ein weiterer Aspekt ist die Wahrung der Vertraulichkeit der Nachrichten. Dabei kann ein Angreifer zwei Ziele verfolgen. Einfach nur Nachrichten entschlüsseln und damit Informationen über den Anwender in Erfahrung bringen oder aber den geheimen Schlüssel in seinen Besitz bringen. Letzteres stellt den maximalen Erfolg eines Angriffes dar, da der Angreifer mit der Kenntnis des Schlüssels dieselben Möglichkeiten, wie der Anwender erlangt. Ein erster Schritt zum Erreichen eines der beiden Ziele könnte die Durchführung eines Replay-Angriffes darstellen. Gegen Replay-Angriffe existiert in WEP kein Schutz, so können aufgezeichnete Nachrichten nochmalig verwendet werden. Gleichzeitig fehlt damit eine Säule zur Durchsetzung der Verbindlichkeit in WLAN-Netzwerken. Ein Beispiel für einen Replay-Angriff könnte folgendes Szenario darstellen. Der Anwender fordert eine Login-Nachricht an, so dass diese zum Anwender übertragen wird. Nach dem Erhalt der Login-Nachricht antwortet der Anwender mit einer Nachricht, die seinen Login-Namen und das dazugehörige Passwort enthält. Zwar werden die Nachrichten verschlüsselt, dennoch könnte ein Angreifer auf deren Inhalt schließen, beispielsweise aufgrund der Nachrichtenlänge. Nach Abmeldung des Anwenders kann sich der Angreifer nun erneut anmelden ohne den Schlüssel zu kennen, einfach durch erneutes Senden der aufgezeichneten Nachrichten. Es gibt zwar eine Sequenznummer in den Nachrichten, diese wird allerdings unverschlüsselt übertragen, so dass sie einfach entsprechend angepasst werden kann.

Zur Verschlüsselung und damit zur Wahrung der Vertraulichkeit von Nachrichten wird der RC4-Stromchiffrierungsalgorithmus in WEP verwendet. Dieser gilt bisher als robust, allerdings gibt es drei Schwächen in dessen Nutzung unter WEP.

1. Wiederverwendung des IV
2. Schlechte Schlüsselwahl
3. Schlüsselbestimmung

Die erste Schwäche ist die Länge des IV. Eigentlich sollte sich der Wert des IVs nicht wiederholen, was bei einer Länge von lediglich 24 Bit nicht gewährleistet werden kann. Weiterhin kann gezeigt werden [FMS01], dass bei einigen Schlüsseln eine unverhältnismäßig hohe Übereinstimmung einiger Bits in den ersten Bytes des mit RC4 erzeugten Zufallsstroms bestehen. Obwohl die Entwickler von RC4 deshalb die Empfehlung geben, die ersten 256 Byte des Zufallsstroms nicht zu nutzen, wird dies in WEP dennoch getan. Als dritte Schwäche hat sich gezeigt, dass die Reihenfolge der Verknüpfung von IV und dem geheimen Schlüssel zu einem Schlüssel, Einfluss auf die Security hat. So ist die Verknüpfung in der, der IV dem geheimen Schlüssel vorangestellt wird ungünstiger als den geheimen Schlüssel dem IV voranzustellen. Aus diesen Schwächen kann ein Angreifer einen Angriff auf den geheimen Schlüssel konstruieren. Da die ersten Bytes beim IEEE 802.11 Standard dem LLC[1] Header

[1] Logical Link Control

entsprechen, kennt ein Angreifer etwas Klartext. Nun wartet der Angreifer bis ein schlechter Schlüssel genutzt wird, welcher durch die Kombination mit dem IV irgendwann auftritt. Damit erhält der Angreifer eine Korrelation zwischen Klartext, Chiffretext und dem Zufallsstrom.

Es gibt nur eine begrenzte Anzahl von Möglichkeiten für das erste Byte des Zufallsstroms die zur Übereinstimmung von Klartext und Chiffretext führen, so dass ein Angreifer nach etwa 60 Nachrichten das erste Byte des Zufallsstroms kennt. Auf diese Weise kann der gesamte Zufallsstrom bestimmt werden und aus ihm schließlich der geheime Schlüssel. Durch die Erhöhung der Schlüssellänge dauert die Bestimmung des geheimen Schlüssels zwar länger, allerdings verhält sich die benötigte Zeit linear zur gewählten Schlüssellänge und nicht exponentiell. Diese Methode zur Bestimmung des geheimen Schlüssels wurde weiter optimiert [TWP07], so dass es einem Angreifer möglich ist, binnen Minuten in ein mit WEP gesichertes WLAN-Netzwerk vollständig einzudringen.

Da TKIP ebenfalls den RC4-Algorithmus zur Verschlüsselung der Nachrichten und damit zur Wahrung der Vertraulichkeit verwendet, ist dieser Angriff prinzipiell mit kleinen Anpassungen auch auf TKIP anwendbar [Wal02a]. Allerdings benötigt ein solcher Angriff auf TKIP verhältnismäßig viel Zeit. Dies ist vornehmlich der Erweiterung des IVs und dem verwendeten Konzept des *Per-Packet Key-Mixings* zu verdanken [EdAr04, Wal02b].

IEEE 802.11i

Im Gegensatz zu WEP sind derzeit keine Designschwächen des IEEE 802.11i Standards bekannt, die sich für einen erfolgreichen Angriff nutzen lassen. Allerdings kann wie bei jedem anderen kryptographischen System, mit Ausnahme von One-Time-Pads [Sch96] ein Brute-Force-Angriff zum Erfolg führen. Bei einem klassischen Brute-Force-Angriff wird jeder mögliche Schlüssel getestet, was je nach Schlüssellänge einen erheblichen Zeitaufwand darstellen kann. Beim IEEE 802.11i Standard beträgt die Länge des PMK 256 Bit, so dass selbst ein Hochleistungsrechner damit nicht in absehbarer Zeit fertig werden würde.

Ein Brute-Force-Angriff kann weiterhin entweder als Online-Angriff oder als Offline-Angriff durchgeführt werden. Bei einem Online-Angriff muss der Angreifer sich während der gesamten Zeit innerhalb der Zelle des anzugreifenden Netzwerks befinden. Damit geht der Angreifer ein hohes Risiko ein, erkannt zu werden und darüber hinaus ist der Angriff aus Sicht der Performance teuer. Für einen Offline-Angriff hingegen benötigt der Angreifer nur einmaligen Zugriff zum Netzwerk, um an verwertbare Informationen für einen lokalen Angriff zu gelangen. Beim IEEE 802.11i Standard werden verwertbare Information während der Authentifizierung mittels dem 4-Wege-Handshake Protokoll ausgetauscht, so dass ein Angreifer die Aufzeichnung einer vollständigen Authentifizierung zwischen Authenticator und Supplicant zum Ziel hat. Ist ein Supplicant bereits an einem Authenticator authentifiziert, kann durch einen

Deauthentifizierungs-Angriff die neue Authentifizierung erzwungen werden. Dazu sendet der Angreifer eine Deauthentifizierungs-Nachricht mit den Informationen des authentifizierten Supplicants an den Authenticator, so dass der Supplicant gezwungen wird sich neu zu authentifizieren.

Nachdem der Angreifer einen vollständigen 4-Wege-Handshake aufgezeichnet hat, kann er die Zelle des Netzwerks verlassen und einen Brute-Force-Angriff auf den PMK durchführen *(Offline Angriff)*. Eine Optimierung des verwendeten Brute-Force-Angriffs stellt die Verwendung eines Wörterbuchs dar. Das Wörterbuch enthält dabei bekannte Schlüssel, die vor dem testen weiterer Kombinationen, getestet werden. Der Erfolg eines solchen Brute-Force-Angriffs hängt vom gewählten PMK ab. Wird der PMK über einen PSK vom Benutzer festgelegt ist die Wahrscheinlichkeit eines erfolgreichen Angriffs höher als bei den Upper-Layer-Authentifizierungsmethoden. Der Anwender wählt häufig einfache Wörter, die auch in Wörterbüchern zu finden sind als statischen PMK, während bei den Upper-Layer-Authentifizierungsmechanismen der PMK, vom gewählten Protokoll und der Sitzung abhängig, dynamisch bestimmt wird.

2.3.5 Security im IEEE P802.11s

Mit dem IEEE P802.11s [IEE08] befindet sich ein weiterer Standard für WLANs, auf Basis des IEEE 802.11 Standards, in der Entstehung. Dieser wird Funktionalitäten für Mesh-Netzwerke spezifizieren, darunter auch ein eigenständiges Security-Konzept. Da sich dieser allerdings noch in einer frühen Phase der Konzeption *(Draft 2.0)* befindet, sind große Teile noch nicht ausreichend spezifiziert, so dass ein Zeitpunkt für die Verabschiedung noch nicht absehbar ist. Zu den offenen Teilen gehören auch einige Teile des Security-Konzepts, so dass im Rahmen dieser Arbeit hierauf nicht zurückgegriffen werden konnte. Dennoch sollen nachfolgend, zur Abrundung des aktuellen Abschnitts einige bedeutende Eckpunkte des Security-Konzepts, welches viele Mechanismen des IEEE 802.11i Standard weiter verwendet, angerissen werden.

Um die offene Authentifizierung zu vermeiden, welche keinerlei Security bietet, und darüber hinaus den im vorangegangenen Abschnitt beschriebenen Offline Angriff auf den PSK bzw. den PMK zu unterbinden wird ein neues Authentifizierungsverfahren mit dem Namen Simultaneous Authentication of Equals *(SAE)* eingeführt. Dieses verwendet ein asymmetrisches Kryptographieverfahren, welches zur Steigerung der Effizienz mit Hilfe von elliptischen Kurven [Kob87, Sch96] realisiert wird. Weiterhin wird eine dreistufige Hierarchie aufgebaut, so dass eine Station neben den Rollen des Authenticators und des Supplicants auch die Rolle eines Schlüsselverteilers einnehmen kann. Damit einher wird die Schlüsselhierarchie des IEEE 802.11i Standards ebenfalls erweitert, so dass für jede Aufgabe wiederum separate Schlüssel zur Verfügung stehen, was der Security zugute kommt.

2.4 Software

Als Basis für die spätere Umsetzung des Konzeptes werden frei verfügbare Implementierungen genutzt. Diese werden nachfolgend vorgestellt.

MadWifi

MadWifi [MW08] ist ein für das Betriebssystem Linux entwickelter WLAN-Treiber für Atheros Chipsätze. Er ist bis auf den Hardware Abstraction Layer *(HAL)* quelloffen, was eine ständige Weiterentwicklung ermöglicht. Zudem wird er von einer Vielzahl von Nutzern verwendet, so dass Neuerungen einer großen Zielgruppe zur Verfügung gestellt werden können und darüber hinaus kann so eine hohe Qualität sichergestellt werden.

Die Implementierung von MadWifi enthält bereits einige Security-Mechanismen des 802.11i Standards. So sind neben der kompletten Unterstützung von WEP, welches im IEEE 802.11i Standard als Deprecated gekennzeichnet ist auch die Verschlüsselungsprotokolle TKIP und CCMP implementiert. In Abhängigkeit von der verwendeten Hardware und Treiberkonfiguration können die in den Verschlüsselungsprotokollen verwendeten Verschlüsselungsalgorithmen RC4 und AES in Hardware oder Software verwendet werden.

WPA/RSN/EAP Authenticator (hostapd)

Der hostapd [MaH08] ist eine von Jouni Malinen entwickelte Software für verschiedene Betriebssysteme, die die Aufgabe des Schlüsselmanagements und der Authentifizierung, nach dem IEEE 802.11i Standard, als Authenticator übernimmt. Sie ist quelloffen und so konzipiert, dass sie als Daemon-Programm im Hintergrund des User Space arbeitet.

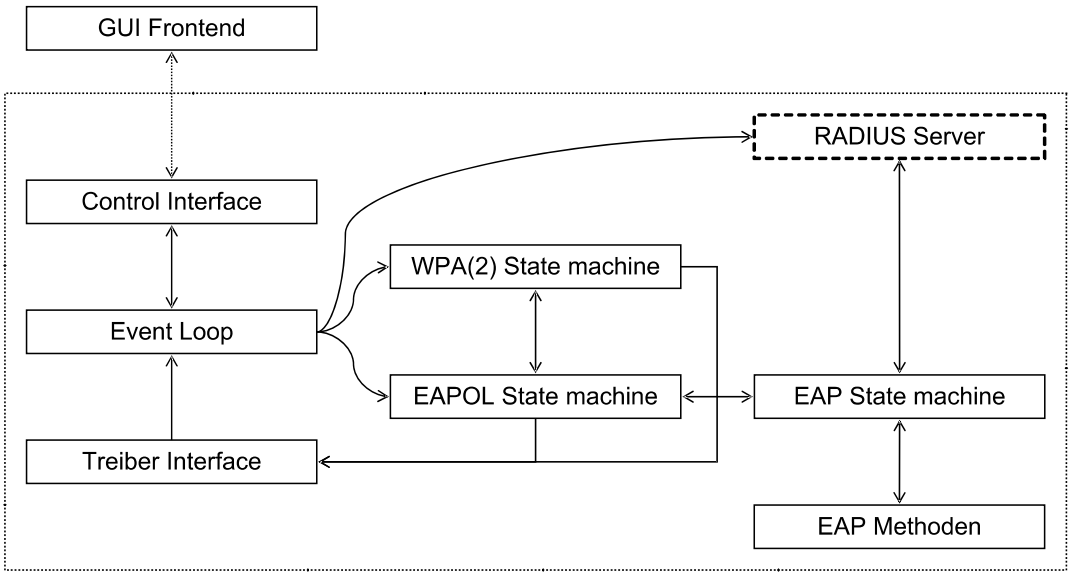

Abbildung 2.5: Struktogramm – hostapd / wpa_supplicant

Um die Wartbarkeit und Weiterentwicklung zu erleichtern ist die gesamte Software Modular aufgebaut. In Abbildung 2.5 sind die wichtigsten Komponenten und deren Zusammenhang untereinander in einem Struktogramm dargestellt. Alle Software-Komponenten, mit Ausnahme des *GUI*[1] *Frontends* gehören zum Kern der Software. Das *GUI Frontend* stellt lediglich eine graphische Nutzeroberfläche zur Konfiguration des *hostapd* zur Verfügung. Somit existiert eine einfache Möglichkeit für den Anwender die Software zu konfigurieren.

Kern des *hostapd* ist die *Event Loop*. Hier wird auf Ereignisse aus dem *Treiber Interface* und mögliche zeitliche Ereignisse reagiert. Die EAP-Zustandsmaschine[2] mit den angebundenen EAP-Methoden, sowie die *RADIUS Server*-Komponente kommen nur bei einer Upper-Layer Authentifizierung zum Einsatz. Welche EAP-Methoden unterstützt werden, sowie welche Treiber Unterstützung finden kann auf diese Weise, ohne eine Änderung an den Kernkomponenten, vorgenommen werden. Bedeutend für die Authentifizierung ist die EAPOL- und WPA / WPA2-Zustandsmaschine. Hier wird der Zustand gesichert, der entscheidet ob auf eine ankommende Nachricht reagiert oder ob eine Nachricht versendet werden muss. Diese Zustandsmaschinen sind bereits im IEEE 802.11i Standard festgelegt und entsprechend in den benannten Komponenten implementiert.

WPA/RSN Supplicant (wpa_supplicant)

Die Funktionalität des Supplicants nach dem IEEE 802.11i Standard übernimmt der wpa_supplicant [MaS08]. Dieser wurde, wie auch der *hostapd*, von Jouni Malinen entwickelt und ist ebenfalls auf diversen Betriebssystemen verwendbar. Er ist quelloffen und arbeitet ebenfalls als Daemon Programm im Hintergrund des User Space.

Er ist ebenfalls in Komponenten unterteilt, die denen des *hostapd* stark ähneln. In Abbildung 2.5 sind die Komponenten des *hostapd* dargestellt. Der Unterschied ist zum einen, dass keine *RADIUS Server*-Komponente[3] existiert und zum anderen die Zustandsmaschinen unterschiedliche Zustände aufweisen können. Die übrigen Komponenten realisieren die Aufgaben, analog denen des *hostapd*. Lediglich in der Implementierung gibt es Detailbedingt Unterschiede.

1 Graphical User Interface
2 Engl.: State machine
3 gestrichelt dargestellt

3 Zugrundeliegendes Konzept

Dieses Kapitel legt das umzusetzende Konzept dar. Dabei lässt sich der Inhalt dieses Kapitels in zwei Teile abgrenzen. Der erste Teil bezieht sich zunächst auf den IEEE 802.11i Standard, im Ad-hoc-Modus. Dazu werden einführend die Mechanismen vorgestellt, die der IEEE 802.11i Standard für den Ad-hoc-Modus vorsieht. Anschließend werden weitere, für die Realisierung des IEEE 802.11i Standards, benötigte Gesichtspunkte betrachtet. Zu ihnen zählen treiberspezifische Funktionalitäten für den IEEE 802.11i Standard, sowie basierend auf der Software *hostapd* und *wpa_supplicant* ein Konzept zur Realisierung der Managementsoftware.

Im zweiten Teil wird auf der Grundlage der im IEEE 802.11i Standard spezifizierten Security-Mechanismen, ein auf die Bedürfnisse von WMNs angepasstes Verfahren entwickelt. Dazu wird die Motivation für ein angepasstes Verfahren dargelegt und die einzelnen Anpassungen gesondert betrachtet. Abschließend werden die Schwächen und Vorzüge des optimierten Verfahrens, gegenüber dem IEEE 802.11i Standard diskutiert.

3.1 IEEE 802.11i Standard in Ad-hoc-Netzwerken

Die in Kapitel zwei beschriebenen Security-Mechanismen beziehen sich im Wesentlichen auf den Infrastruktur-Modus von WLANs. Für den Ad-hoc-Modus müssen daher einige dieser Mechanismen, den Gegebenheiten von Ad-hoc-Netzwerken angepasst werden. So existiert kein AP als zentrale Instanz, der die Zelle definiert, sowie hauptverantwortlich für die Umsetzung der Security-Mechanismen ist. Nachfolgend werden die nötigen Anpassungen, an die neuen Gegebenheiten, der verschiedenen Security-Mechanismen des IEEE 802.11i Standards betrachtet.

Authentifizierung

Anders als in Infrastruktur-Netzwerken kann in einem Ad-hoc-Netzwerk potenziell jede Station mit jeder anderen, in Reichweite befindlichen, Station kommunizieren. Damit muss sich jede Station mit jeder anderen, in Reichweite befindlichen, Station gleichermaßen authentifizieren. Die Authentifizierung initiiert, nach dem IEEE 802.11i Standard, der Authenticator, während der Supplicant sich authentifizieren muss. Somit nimmt eine Station im Ad-hoc-Modus sowohl die Rolle eines Authenticators, als auch eines Supplicants ein. Dies ist in Abbildung 3.1 an einem Beispiel von zwei Stationen dargestellt.

Da das 4-Wege-Handshake Protokoll eine gegenseitige Authentifizierung zwischen zwei Stationen gewährleistet, ist es aus Sicht der Security nicht notwendig, dass das 4-Wege-Handshake Protokoll einmal von Station 1 und einmal von Station 2 initiiert wird. Durch die redundante Ausführung des 4-Wege-Handshake Protokolls ergibt sich allerdings der Vorteil, dass die Zustandsmaschine des Authenticators nicht zwischen Infrastruktur-Modus und Ad-hoc-Modus unterscheiden muss. Erst wenn beide 4-Wege-

Handshakes erfolgreich durchgeführt wurden, darf nach dem IEEE 802.1X Standard die Verbindung für die paarweise Kommunikation freigegeben werden.

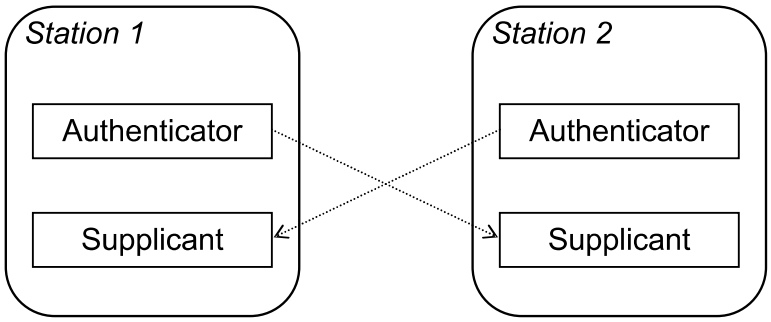

Abbildung 3.1: Authentifizierung im Ad-hoc-Modus

Mit der redundanten Ausführung des 4-Wege-Handshakes, im Ad-hoc-Modus, steigt allerdings auch die Belastung des Mediums. Abbildung 3.2 verdeutlicht dies an einem Beispiel von drei Stationen, die über volle Konnektivität zueinander verfügen.

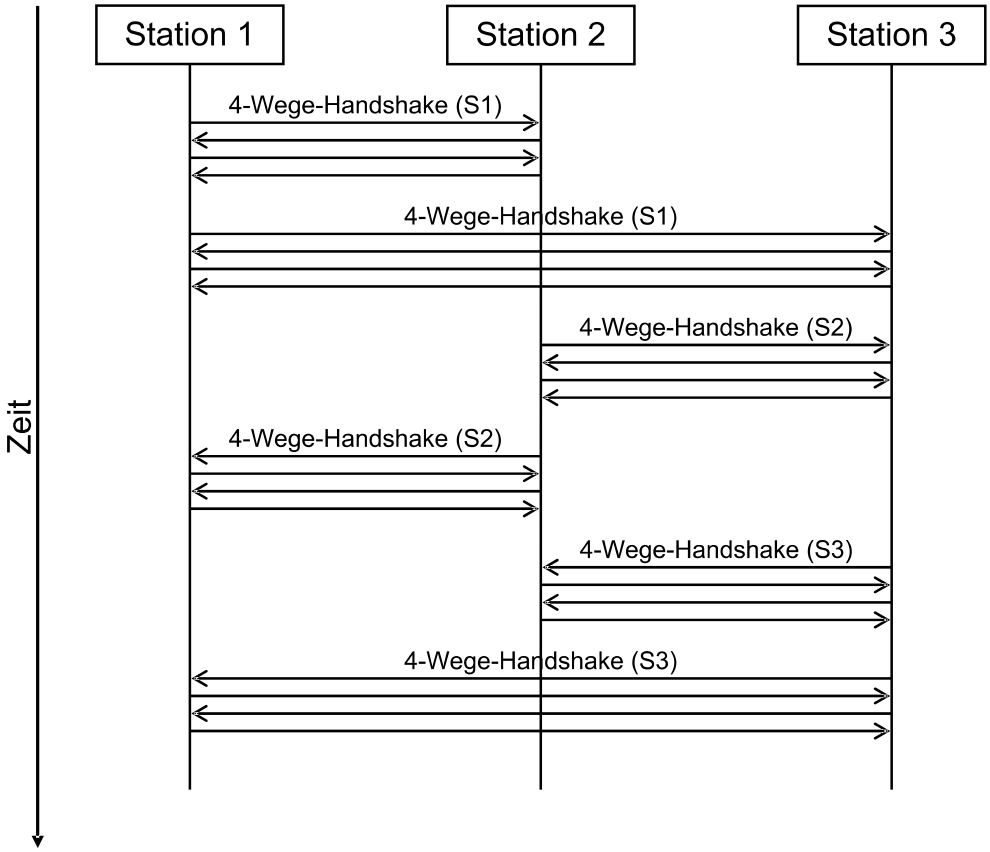

Abbildung 3.2: 4-Wege-Handshake im Ad-hoc-Modus

Bei der Authentifizierung initiiert Station 1 einen 4-Wege-Handshake mit Station 2, sowie mit Station 3. Station 2 initiiert ihrerseits einen 4-Wege-Handshake mit Station 3, sowie mit Station 1 und Station 3 letztendlich mit Station 2 und Station 1. Würde sich zusätzlich eine vierte Station in Reichweite der drei Stationen befinden, so müssten sich die drei Stationen mit der vierten authentifizieren und die vierte ihrerseits mit den drei

Stationen. Daraus lässt sich für die Authentifizierung von *n*-Stationen, bei vollständiger Konnektivität *(Worst Case)* innerhalb des Ad-hoc-Netzwerks, folgender Kommunikationsaufwand bestimmen.

$$O(n * (n - 1)) = O(n^2 - n) = \underline{\underline{O(n^2)}}$$

Jede Station muss im Falle der vollen Konnektivität den 4-Wege-Handshake *(n - 1)* mal initiieren. Dies bedeutet, dass mit einer zunehmenden Anzahl von Stationen, die für die Authentifizierung, benötigte Belastung des Mediums quadratisch ansteigt.

Integrität, Vertraulichkeit und Verbindlichkeit

Die Integrität der Frames im Ad-hoc-Modus kann analog zum Infrastruktur-Modus durch Michael für das Verschlüsselungsprotokoll TKIP und durch den CBC Message Authentication Code für das Verschlüsselungsprotokoll CCMP gewährleistet. Allerdings sieht der IEEE 802.11i Standard lediglich die Verwendung des Verschlüsselungsprotokolls CCMP im Ad-hoc-Modus vor, so dass TKIP nicht Standardkonform ist und daher nur am Rande Erwähnung findet.

Zur Gewährleistung der Vertraulichkeit der Informationen in den Frames, wird das Verschlüsselungsprotokoll CCMP unverändert aus dem Infrastruktur-Modus übernommen. Auch für den Ad-hoc-Modus definiert der IEEE 802.11i Standard nur einen Mechanismus gegen Replay-Angriffe, sowie einen Mechanismus zur Überprüfung der Absenderauthentizität für die paarweise Kommunikation. Damit gibt es keinen Mechanismus zur Gewährleistung der Verbindlichkeit von Frames.

Schlüsselmanagement

Die für die Sicherung der Integrität und Vertraulichkeit von Frames benötigten Schlüssel werden, analog zum Infrastruktur-Modus, über zwei Schlüsselhierarchien erzeugt. Allerdings wird der 4-Wege-Handshake zur Authentifizierung zwischen paarweisen Stationen einmal von jeder Station initiiert. Dies bedeutet, dass auch zwei PTKs erzeugt werden. Da die Kommunikationspartner allerdings nur einen PTK zur Sicherung der Vertraulichkeit und Integrität ihrer Kommunikationsbeziehung benötigten, sieht der IEEE 802.11i Standard nur die Nutzung eines PTKs vor. So wird festgelegt, dass der vom Authenticator mit der höheren MAC-Adresse initiierte 4-Wege-Handshake dazu genutzt wird, den für die Kommunikation benötigten PTK zu erzeugen. Da im Ad-hoc-Modus jede Station ebenfalls die Rolle eines Authenticators einnimmt, erzeugt jede Station zudem, analog dem AP in Infrastruktur-Netzwerken, einen GMK. Aus diesem wird dann, innerhalb der Schlüsselhierarchie, durch Ableitung ein GTK zur Sicherung einer Multicast-Verbindung erzeugt. Anschließend wird der GTK, unter Nutzung des Group Key-Handshakes, an die authentifizierten Supplicants verteilt.

In der Zusammenfassung ergibt sich so, unter Verwendung der unveränderten Schlüsselhierarchien, ein verändertes Bild in der Nutzung der Schlüssel. Tabelle 3.1

stellt hierzu, für eine Zelle innerhalb der alle Stationen volle Konnektivität zueinander haben, die benötigten Schlüssel, sowie deren Verwendung für den Infrastuktur-Modus und den Ad-hoc-Modus gegenüber.

	Infrastruktur-Modus	**Ad-hoc-Modus**
AP	- ***n***[1] x PTK (senden, empfangen) - ***1*** x GTK (senden)	
Station	- ***1*** x PTK (senden, empfangen) - ***1*** x GTK (empfangen)	- ***n***[2] x PTK (senden, empfangen) - ***1*** x GTK (senden) - ***n*** x GTK (empfangen)

Tabelle 3.1: Schlüsselverwendung im Infrastruktur- und Ad-hoc-Modus

Die Auswahl des für die Entschlüsselung eines Frames zu nutzenden Schlüssels, muss der Empfangsstation, durch die Senderstation, mitgeteilt werden. So kann der bereits im Infrastruktur-Modus vom AP verwendete Schlüsselcache, zur Verwaltung der PTKs verwendet werden. Für den GTK ist zunächst, analog zum Infrastruktur-Modus, die Verwendung einer Schlüsseltabelle vorgesehen, die über ein 2 Bit Schlüsselindex, eingebettet im Frame, indexiert wird.

3.2 Ereignisverarbeitung im Treiber für den IEEE 802.11i Standard

Im Infrastruktur-Modus erfolgt die Zuordnung zwischen dem Supplicant und dem Authenticator explizit. Der Supplicant sucht zunächst mittels passiven oder aktiven Scanning [NMG01] einen Authenticator, um sich anschließend durch eine Offene Authentifizierung bekannt zu machen und bei diesem Authenticator anzumelden. Wurde die Anmeldung *(Assoziierung)* vom Authenticator bestätigt, können beide Parteien miteinander kommunizieren. Diese Prozedur ist sowohl für eine Kommunikation ohne Verwendung eines Security-Verfahrens, wie auch mit einem Security-Verfahren notwendig.

Der Empfang der Assoziierungs-Anforderung auf der Seite des Authenticators löst innerhalb des Treibers ein Ereignis aus. Dieses kann entweder durch den Treiber selbst verarbeitet oder für andere Applikationen zur Verfügung gestellt werden. Speziell die Auslagerung der Authentifizierung, nach dem IEEE 802.1X Standard, und des für die Umsetzung des IEEE 802.11i Standards benötigten Schlüsselmanagements in eine externe Software verlangt nach einer Schnittstelle, seitens des Treibers, zur Kommunikation, sowie zur Mitteilung von Ereignissen. Die Auslagerung der Authentifizierung und des Schlüsselmanagements in eine externe Software hat die

[1] Anzahl der Stationen
[2] Anzahl der Nachbarstationen

Vorteile, dass die Wartbarkeit der einzelnen Komponenten erleichtert, die Flexibilität in Hinblick auf die Portierbarkeit erhöht, sowie die Umsetzung zukünftiger Erweiterungen beschleunigt wird. So sehen bereits diverse Treiber eine solche Schnittstelle vor, um unter Nutzung der zur Verfügung stehenden Software *hostapd* und *wpa_supplicant* die Authentifizierung und das Schlüsselmanagement für den Infrastruktur-Modus zu realisieren. Abbildung 3.3 zeigt das hierfür vom Treiber umzusetzende Ereignismodell. Dieses Ereignismodell bildet ebenfalls die Grundlage zur Entwicklung einer *Managementsoftware*, zur Umsetzung des IEEE 802.11i Standards für den Ad-hoc-Modus.

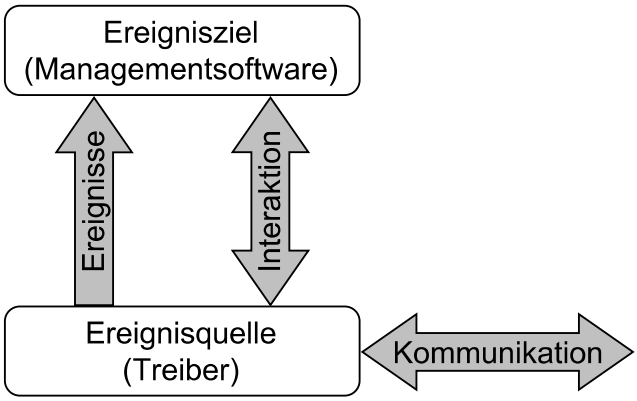

Abbildung 3.3: Konzept-Struktogramm – Ereignismodell

Die Verwendung eines Ereignismodells zur Kommunikation zwischen dem Treiber und einer externen Software hat darüber hinaus den weiteren Vorteil, dass die vorhandenen Systemressourcen so schonend genutzt werden. Neben dem *New Station*-Ereignis, welches durch die Assoziierung des Supplicants beim Authenticator ausgelöst wird, muss ein *Station Leave*-Ereignis beim Verlassen des Supplicants ausgelöst werden. Dies wird in analoger Weise zur Assoziierungs-Anforderung durch eine explizite Deassoziierungs-Anforderung, durch den Supplicant, vorgenommen.

Um den Treiber konfigurieren zu können, beispielsweise zur Festlegung des verwendeten Verschlüsselungsprotokolls, welches aus Gründen der Performance im Treiber realisiert werden sollte, benötigt der Treiber eine weitere Schnittstelle. Diese Schnittstelle erlaubt es, neben der Initialen Konfiguration, Frames zwischen Treiber und der externen Software auszutauschen.

Im Ad-hoc-Modus werden die Mechanismen zur Authentifizierung und zum Schlüsselmanagement, analog zum Infrastruktur-Modus, in eine externe Software, der *Managementsoftware*, realisiert. Zur Kommunikation mit der *Managementsoftware* verwendet der Treiber ebenfalls das in Abbildung 3.3 dargestellte Ereignismodell des Infrastruktur-Modus. Allerdings bildet der Authenticator im Ad-hoc-Modus, im Gegensatz zum Infrastruktur-Modus keine zentrale Instanz. Damit ist eine explizite Assoziierung zwischen Authenticator und Supplicant in der bekannten Form nicht möglich, so dass der Treiber die notwendige Erzeugung der Ereignisse implizit über die *Beacon*-Frames vornehmen muss. In der Abbildung 3.4 lassen sich die möglichen

Situationen für die Erzeugung von Ereignissen im Ad-hoc-Modus aufzeigen. Dabei wird davon ausgegangen, dass sämtliche Stationen innerhalb der Zellen über uneingeschränkte Konnektivität zueinander verfügen.

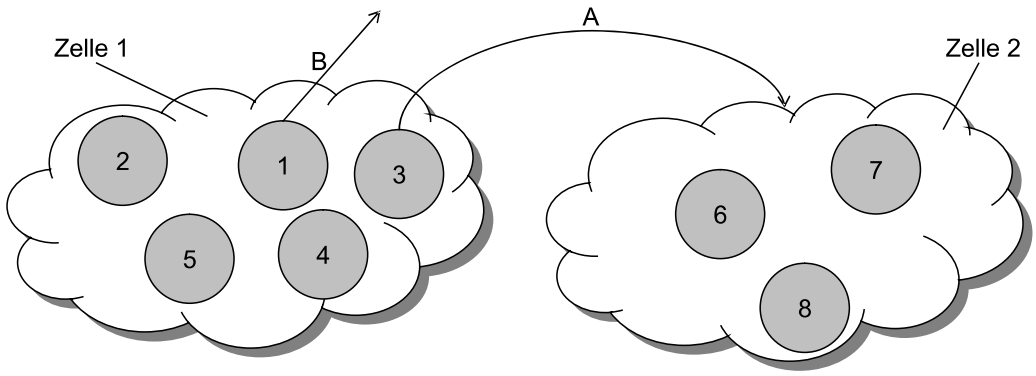

Abbildung 3.4: Mögliche Verteilung von Stationen im Ad-hoc-Modus

Zunächst ist ein denkbares Szenario A, dass eine Station, in diesem Fall die Station 3 ihre ursprüngliche Zelle 1 verlässt und in eine andere Zelle 2 wechselt. In diesem Fall ergeben sich drei Änderungen, die Ereignisse nach sich ziehen. Zunächst ergibt sich eine Änderung innerhalb der Zelle 1. So befindet sich Station 3 außerhalb der Zelle 1, so dass die Treiber der Stationen 1, 2, 4 und 5 jeweils ein *Station Leave*-Ereignis, für die Station 3 erzeugen müssen. Stattdessen befindet sich die Station 3 nun in der Zelle 2. Dies bedeutet für die Treiber der Stationen 6, 7 und 8, innerhalb der Zelle 2, dass sie jeweils ein *New Station*-Ereignis, für die hinzugekommene Station 3, auslösen müssen. Der Treiber der Station 3, muss seinerseits die drei neuen Stationen 6, 7 und 8 jeweils über ein *New Station*-Ereignis melden. Dabei spielt es keine Rolle, ob die Station 3 in der Vergangenheit bereits zu einer anderen Zelle gehört hat oder nicht.

Ein weiteres Szenario B entstünde, wenn die Station 1, beispielsweise aufgrund von Mobilität, ihre Konnektivität zur Zelle 1 verliert. In diesem Szenario müssen die Treiber der Stationen 2, 3, 4 und 5 jeweils ein *Station Leave*-Ereignis erzeugen. Damit sind alle möglichen Vorgänge innerhalb von Ad-hoc-Netzwerken abgedeckt, die eine Erzeugung von Ereignissen nach sich ziehen müssen. Tabelle 3.2 stellt noch einmal zusammenfassend die vier möglichen Ereignisse dar.

Szenario	Beschreibung	Ereignis
A	- Betrachtete Station verbindet sich mit einer Zelle (beinhaltet Wechseln der Zelle)	- New Station
	- Andere Station verbindet sich mit betrachteter Station	- New Station
	- Andere Station wechselt zu einer von der betrachteten Station verschiedenen Zelle	- Station Leave
B	- Andere Station verlässt Reichweite der betrachteten Station	- Station Leave

Tabelle 3.2: Überblick der möglichen Ereignisse im Ad-hoc-Modus

3.3 Managementsoftware für den IEEE 802.11i Standard

3.3.1 Authentifizierung

Die paarweise Authentifizierung, im Ad-hoc-Modus, zwischen den Stationen erfolgt auf Basis des PMKs. Nachdem eine Assoziierung durch den Treiber signalisiert wurde oder eine erfolgreiche Authentifizierung eine längere Zeit zurückliegt *(Re-Authentifizierung)*, beginnt die Managementsoftware mit dem 4-Wege-Handshake Protokoll. Der Ablauf des 4-Wege-Handshake Protokolls, wird nach dem IEEE 802.11i Standard und dem IEEE 802.1X Standard durch die Spezifizierung von Zustandsmaschinen geregelt. Diese Zustandsmaschinen werden jeweils für den Authenticator und den Supplicant separat in die Managementsoftware integriert.

Da der IEEE 802.11i Standard, im Ad-hoc-Modus, die Initiierung des 4-Wege-Handshake Protokolls durch beide Kommunikationspartner vorsieht, um die Zustandsmaschinen des Authenticators unverändert aus dem Infrastruktur-Modus übernehmen zu können, können sich Probleme ergeben. Beispielsweise kann es vorkommen, dass der Authenticator der Station 1 bereits mit der Authentifizierung beginnt, während der Authenticator der Station 2 noch kein Assoziierungs-Ereignis erhalten hat. Daher wird der Empfang der ersten Nachricht des 4-Wege-Handshakes ebenfalls als Assoziierungs-Ereignis, in der Managementsoftware, aufgefasst und der 4-Wege-Handshake initiiert. Aufgrund der Nutzung eines gemeinsamen Mediums, kann darüber hinaus ein weiteres Problem während des 4-Wege-Handshakes auftreten. Es kann vorkommen, dass versendete Nachrichten ihren Empfänger nicht erreichen, was zu einem erneuten versenden der entsprechenden Nachricht führt. Da der IEEE 802.11i Standard jedoch einen zeitlichen Rahmen für die Authentifizierung vorsieht, könnte ein 4-Wege-Handshake erfolgreich durchgeführt wurden sein, während der zweite 4-Wege-Handshake nicht erfolgreich verlief. Die paarweise Authentifizierung im Ad-hoc-Modus, nach dem IEEE 802.11i Standard, gilt aber nur dann als erfolgreich, wenn beide 4-Wege-Handshakes erfolgreich durchgeführt wurden. Tritt diese Situation auf, so sieht der IEEE 802.11i Standard eine Wiederholung der kompletten Authentifizierung vor, spezifiziert aber nicht wann diese Wiederholung durchzuführen ist.

Denkbar für eine Wiederholung sind zwei verschiedene Strategien. Eine Strategie wäre die unmittelbare Wiederholung der Authentifizierung, solang bis die Authentifizierung erfolgreich verlaufen ist. Diese Strategie verspricht, mit hoher Wahrscheinlichkeit, zwischen Stationen mit statischen Positionen erfolgreich zu sein, wodurch die benötigte Zeit zur Authentifizierung minimiert wird. Allerdings wird das Medium, insbesondere bei Stationen mit dynamischen Positionen, durch das ständige Versenden der Nachrichten stark belastet. Daher wäre es sinnvoll die Authentifizierung erst nach einer gewissen Zeitspanne zu wiederholen. Damit wäre das Medium auch bei mehrfachem Fehlschlagen der Authentifizierung minimal belastet, jedoch würde eine Erhöhung der Zeit zur Authentifizierung damit einhergehen.

Um aus beiden Strategien möglichst die Vorzüge zu nutzen, wird eine kombinierte Lösung verwendet. Schlägt eine Authentifizierung fehl, wird sie nach kurzer Zeit

wiederholt. Bei jedem weiteren Fehlschlagen wird die Zeit, zwischen einer Wiederholung der Authentifizierung, Schrittweise bis zu einer zeitlichen Obergrenze erhöht. Somit ergibt sich eine Strategie, die die Belastung des Mediums gering hält und dennoch eine rasche Authentifizierung anstrebt.

3.3.2 Schlüsselmanagement

Neben der Authentifizierung übernimmt die Managementsoftware die Aufgabe des Schlüsselmanagements, für den IEEE 802.11i Standard im Ad-hoc-Modus. Das Schlüsselmanagement beinhaltet zunächst das Verwalten, Erzeugen und Aktualisieren des GMKs, die Verwaltung des PMKs und die Ableitung der zur Sicherung der Verbindungen benötigten Schlüssel GTK und PTK. Darüber hinaus müssen die für die Sicherung von EAPOL-Nachrichten benötigten Schlüssel verwaltet werden.

Um eine mögliche Vorhersagbarkeit der erzeugten GMKs und Nonce-Werte auszuschließen müssen kryptographisch starke Zufallsfunktionen verwendet werden. Auf modernen Betriebssystemen, wie Microsoft Windows, Linux oder Mac OS X existieren jedoch hierfür ausreichende Mechanismen, die in der späteren Implementierung verwendet werden könnten. Der PMK wird in der entwickelten Managementsoftware durch den Anwender, über einen PSK, festgelegt. Eine zukünftige Erweiterung um Upper-Layer Authentifizierungsmethoden ist im entwickelten Softwaredesign *(siehe Abschnitt 3.3.4)* bereits vorgesehen, wird aber im Rahmen dieser Arbeit nicht weiter verfolgt.

Der IEEE 802.11i Standard spezifiziert die Ableitung von PTK und GTK, so dass diese unverändert in die Managementsoftware übernommen werden. Auch die Situationen für eine Aktualisierung der Schlüssel beschreibt der IEEE 802.11i Standard. So muss der Authenticator der Managementsoftware den GTK in gewissen Zeitabständen oder bei einer Deauthentifizierung aktualisieren und verteilen. Neben dem GTK wird auch der PTK nach Ablauf einer gewissen Zeitspanne erneut ausgehandelt.

3.3.3 Deauthentifizierung

Für die Deauthentifizierung von Stationen, im Infrastruktur-Modus, sieht der IEEE 802.11i Standard zwei Möglichkeiten vor. Diese werden unverändert in den Ad-hoc-Modus übernommen. Die erste Möglichkeit besteht in einer expliziten Deauthentifizierung, unter Nutzung eines speziellen Management-Frames. Dieses enthält, die für eine Deauthentifizierung notwendige MAC-Adresse, der zu deauthentifizierenden Station, und zusätzlich alle MAC-Adressen, der mit dieser Station authentifizierten Stationen. Nach dem Empfang eines solchen Frames wird die entsprechende Station deauthentifiziert und sämtliche GTKs, die der Station bekannt waren, werden erneuert.

Es kann allerdings vorkommen, dass eine Station ein solches Management-Frame nicht senden kann, beispielsweise weil ein technischer Defekt vorliegt oder die Station den Bereich der Zelle verlassen hat. In diesem Fall empfangen die Treiber der übrigen Stationen keine Beacon-Frames, von der entsprechenden Station. Werden über einen längeren Zeitraum keine Beacon-Frames, von einer im Treiber assoziierten Station empfangen, kann davon ausgegangen werden, dass die Station nicht länger zur Zelle gehört. Dies bekommt die Managementsoftware, vom Treiber über ein *Station Leave*-Ereignis mitgeteilt, so dass sie die Deauthentifizierung der entsprechenden Station vernehmen kann. Abschließend müssen auch in diesem Fall sämtliche GTKs, die der Station bekannt waren, erneuert werden.

3.3.4 Softwaredesign

Da der IEEE 802.11i Standard eine Adaptierung der Security-Mechanismen aus dem Infrastruktur-Modus auf den Ad-hoc-Modus vorsieht, empfiehlt es sich vorhandene Software-Komponenten des Infrastruktur-Modus, als Basis für die Realisierung der Managementsoftware zu nutzen. Mit den beiden Applikationen *hostapd* und *wpa_supplicant* liegt bereits eine funktionstüchtige Umsetzung des IEEE 802.11i Standards für den Infrastruktur-Modus vor.

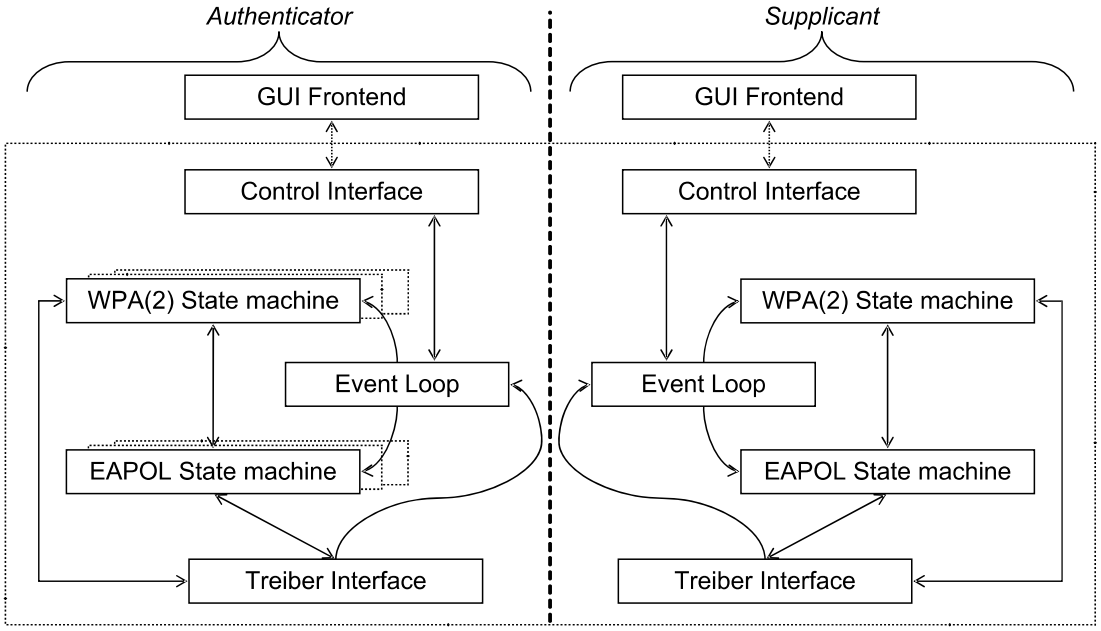

Abbildung 3.5: Struktogramm – Ausgangssituation der Managementsoftware

Wie in *Abschnitt 3.1* beschrieben, stellt jede Station im Ad-hoc-Modus einen Authenticator und einen Supplicant dar. Da der *hostapd* den Authenticator und der *wpa_supplicant* den Supplicant im Infrastruktur-Modus realisieren, werden diese im Hinblick auf die Verkürzung der Entwicklungszeit als Ausgangspunkt für die Entwicklung der Managementsoftware genutzt. Die Verschmelzung von *hostapd* und *wpa_supplicant*, zu einer einzigen Managementsoftware, erleichtert die Verwaltung der

Gegenstellen, sowie die notwendige Kommunikation zwischen Authenticator und Supplicant. Eine Kommunikation zwischen Authenticator und Supplicant ist beispielsweise beim Schlüsselmanagement[1] und der Überprüfung der erfolgreichen Authentifizierung[2] erforderlich.

Die Ausgangssituation der Managementsoftware ist in Abbildung 3.5 dargestellt. Da die Managementsoftware den Fokus auf die Verwendung eines PSK als PMK legt, werden die Komponenten *Radius Server*, *EAP State machine* und *EAP Methoden* aus Abbildung 2.4, welche für die Upper-Layer Authentifizierung benötigt werden, in der Abbildung nicht berücksichtigt. Nachfolgend wird auf dieser Grundlage, schrittweise, die Managementsoftware, wie sie schließlich in Abbildung 3.7 dargestellt ist, aufgebaut.

Im Ad-hoc-Modus muss der Supplicant keinen Authenticator durch Scanning auswählen, da jede andere Station in der Zelle einen Authenticator darstellt. Damit kann der Supplicant nunmehr eine passive Aufgabe einnehmen. Die Gegenstelle und damit den Authenticator wählt der Treiber über eine implizite Assoziierung aus und teilt diese Auswahl, mit Hilfe eines *New Station-Ereignis*, mit. Da eine Assoziierung im Ad-hoc-Modus sowohl die Initiierung einer Authentifizierung mit der Gegenstelle, als auch eine Authentifizierung durch die Gegenstelle nach sich zieht, müssen für jede Gegenstelle separate Zustandsmaschinen für den Authenticator und den Supplicant zur Verfügung gestellt werden. Die im IEEE 802.11i Standard spezifizierte Zustandsmaschine, zur Realisierung des 4-Wege-Handshakes, sowie des Group Key-Handshakes wird in der Komponente *WPA(2) State machine* umgesetzt. Dabei wird eine Unterscheidung zwischen Authenticator und Supplicant vorgenommen, wie es der Standard vorsieht. Zusätzlich bedient sich der IEEE 802.11i Standard einer Zustandsmaschine des IEEE 802.1X Standards, zur Realisierung der Authentifizierung. Diese wird in der Komponente *EAPOL State machine*, separat für den Authenticator und den Supplicant realisiert. Zusammengefasst werden entsprechend, zur Umsetzung der Authentifizierung, für den Authenticator zwei Zustandsmaschinen, fortan als *Authenticator-Instanz* bezeichnet und für den Supplicant zwei Zustandsmaschinen, fortan als *Supplicant-Instanz* bezeichnet benötigt.

Da eine Station für jede Gegenstelle sowohl eine Authenticator-Instanz als auch eine Supplicant-Instanz benötigt, muss eine Datenstruktur zur Verwaltung selbiger umgesetzt werden. Im Infrastruktur-Modus muss der Authenticator per Definition mehrere Supplicants verwalten können, so dass hierfür bereits eine Datenstruktur zur Verwaltung der Authenticator-Instanzen zur Verfügung steht. Im Gegensatz zum Authenticator kann der Supplicant sich im Infrastruktur-Modus nur explizit mit einem Authenticator authentifizieren, so dass er keine Datenstruktur zur Verwaltung von Supplicant-Instanzen vorsieht. Aus diesem Grund bietet es sich an, die bereits vorhandene Datenstruktur des Authenticators so zu erweitern, dass sie neben den Authenticator-Instanzen, auch Supplicant-Instanzen verwalten kann. Um eine einfache

[1] PTK des vom Authenticator mit der höheren MAC-Adresse initiierten 4-Wege-Handshakes, wird zur Sicherung der Unicast-Verbindung verwendet.

[2] 4-Wege-Handshake muss in beide Richtungen erfolgreich durchgeführt werden.

Verwaltung der Datenstruktur zu ermöglichen wird festgelegt, dass das *Treiber Interface* des Authenticators auf die Ereignisse des Treibers reagiert. Damit würde das *Treiber Interface* des Supplicants nur noch Authentifizierungs-Nachrichten entgegen nehmen. Da der Treiber ohnehin keine Unterscheidung zwischen Authentifizierungs-Nachrichten für den Authenticator und Authentifizierungs-Nachrichten für den Supplicant vornimmt, würden beide *Treiber Interfaces* alle Authentifizierungs-Nachrichten erhalten. Somit genügt für diese Aufgabe ebenfalls ein *Treiber Interface*, so dass das *Treiber Interface* des Supplicants in der Managementsoftware nicht länger benötigt wird.

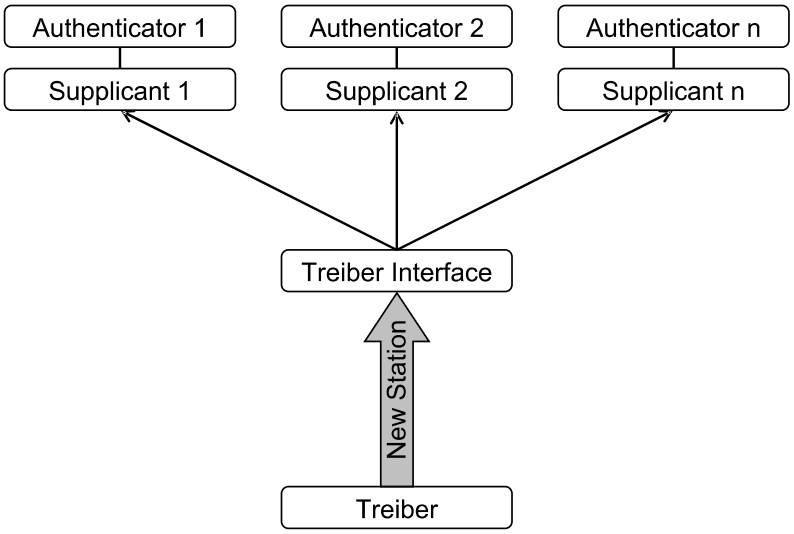

Abbildung 3.6: Erzeugung der Authenticator- und Supplicant-Instanzen

Kommt es zur Auslösung eines *New Station*-Ereignisses, durch den Treiber, so wird wie in Abbildung 3.6 dargestellt sowohl eine Authenticator-Instanz als auch eine Supplicant-Instanz im *Treiber Interface* erzeugt und registriert. Eine Registrierung der Instanzen ist notwendig, um die später eingehenden Authentifizierungs-Nachrichten der verantwortlichen Instanz zuzuordnen.

Um auf die Ereignisse des *Treiber Interfaces* zu reagieren, wird ebenfalls eine *Event Loop* verwendet. Selbige bildet, wie in Abbildung 3.7 dargestellt, analog zum *hostapd* und zum *Supplicant*, die Zentralkomponente der Managementsoftware. Neben den, von dem *Treiber Interface* eingehenden Ereignissen, können so auch zeitabhängig Ereignisse von beiden Instanzen zentral verwaltet werden. Liegt ein Ereignis an, so sendet die *Event Loop* dies, unter Berücksichtigung des zugehörigen Kontextes, an die für die Bearbeitung zuständige Zustandsmaschine. Diese ändern entsprechend ihren Zustand und verarbeiten das Ereignis, gemäß der Spezifikation des IEEE 802.11i Standards bzw. des IEEE 802.1X Standards.

Für die einfache Konfiguration werden weiterhin zwei *Control Interfaces* zur Verfügung gestellt. So können die bereits entwickelten GUIs des *hostapd* und des *wpa_supplicant* ebenfalls verwendet werden. Allerdings genügt zur Konfiguration der Managementsoftware im Prinzip bereits die GUI des *hostapd*.

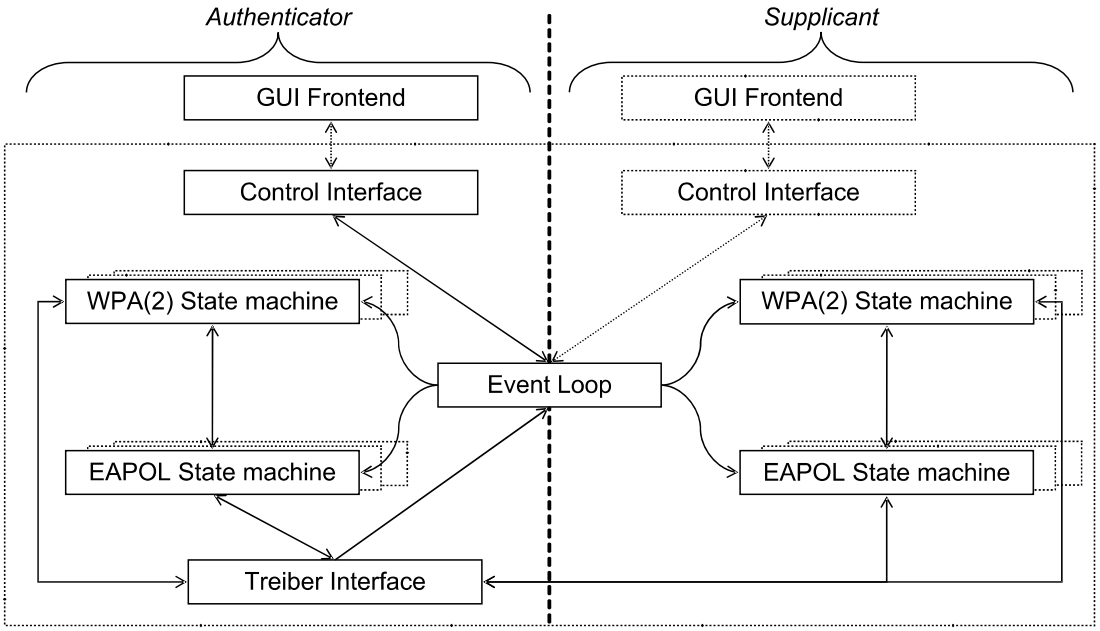

Abbildung 3.7: Struktogramm – Managementsoftware

3.4 Angepasstes Verfahren für WMNs

3.4.1 Motivation

Wie bereits in *Abschnitt 2.1* beschrieben, nutzen WMNs ein Multihop-Routing-Protokoll um einen Nachrichtenaustausch, auch zwischen nicht direkt erreichbaren Stationen, zu ermöglichen. Dies bedeutet, dass einige Stationen *(Router-Stationen)* sich in den Dienst anderer Stationen stellen müssen und so als Vermittler fungieren. Mit der Nutzung von Router-Stationen ändert sich gegenüber Ad-hoc-Netzwerken ein, auch für die Security, bedeutender Punkt. So gibt es im Allgemeinen in WMNs keine Punkt-zu-Punkt-Verbindungen *(Unicast-Verbindungen)*, sondern lediglich Ende-zu-Ende-Verbindungen zwischen den Kommunikationspartnern. Der IEEE 802.11i Standard für Ad-hoc-Netzwerke sieht jedoch, wie in Abbildung 3.8 dargestellt, eine individuelle Sicherung jeder Unicast-Verbindung vor.

Möchte beispielsweise Station 1 mit Station 3 kommunizieren, so fungieren Station 4 und 5 als Router-Stationen. Da somit eine Kommunikation auf die Kooperation anderer Stationen angewiesen ist, kann die Vertraulichkeit und Integrität zwischen den Kommunikationspartnern nur gewahrt werden, wenn die Router-Stationen vertrauenswürdig sind. Damit ist es für die Security unerheblich, ob eine paarweise Sicherung der Vertraulichkeit und Integrität, wie es der IEEE 802.11i Standard für Ad-hoc-Netzwerke vorsieht, gewährleistet wird oder ob für das gesamte WMN eine solche Vorkehrung besteht. Aus Sicht der Performance des Schlüsselmanagements in WMNs ist dies allerdings ein gravierender Unterschied. So würde beispielsweise die Aushandlung und Verteilung von weniger Schlüsseln eine geringere Bandbreite

beanspruchen, was dann einen höheren Datendurchsatz von Nutzdaten, speziell während der Schlüsselaktualisierung, innerhalb des Netzwerks erlauben würde.

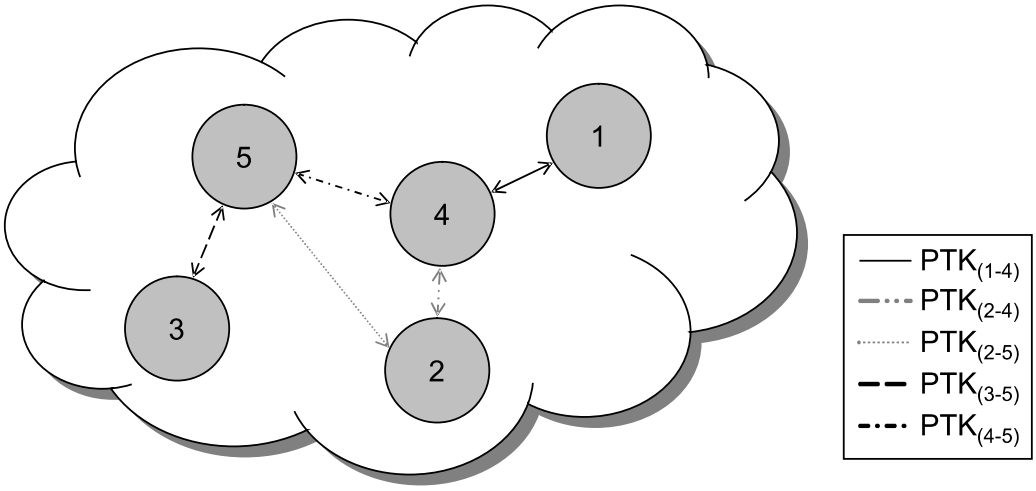

Abbildung 3.8: Sicherung der Verbindungen nach dem IEEE 802.11i Standard

Unter dem Gesichtspunkt, dass sämtliche Stationen innerhalb des WMNs vertrauenswürdig und damit authentifiziert sein müssen, sowie im Hinblick auf die Möglichkeit einer Vereinfachung des Schlüsselmanagements, könnte auch die Authentifizierung vereinfacht werden. So haben die paarweisen 4-Wege-Handshakes, neben der Authentifizierung beider beteiligter Stationen, die Aufgabe, die Schlüssel zur Sicherung der Unicast-Verbindung auszuhandeln. Indem Unicast-Verbindungen in WMNs nicht gesondert gesichert werden müssen und alle Stationen innerhalb des WMNs als authentifiziert anzusehen sind, würde eine Authentifizierung zwischen einer Station des WMNs und der nicht authentifizierten Station genügen. Dies würde der Belastung des Mediums und damit dem Datendurchsatz zugute kommen, sowie darüber hinaus in vielen Fällen die Initiale Anlaufzeit[1] des WMNs verkürzen. Da die Initiierung, nach dem IEEE 802.11i Standard, des 4-Wege-Handshakes für paarweise Stationen das Medium doppelt belastet, ohne einen Mehrgewinn an Security zu erzielen liegt an dieser Stelle ebenfalls eine Optimierungsmöglichkeit.

Ein weiteres bisher nur am Rande erwähntes Szenario stellt die Mobilität von Stationen dar. So muss sich eine mobile Station, nach dem IEEE 802.11i Standard, mit jeder neuen Nachbarstation innerhalb des WMNs erneut authentifizieren. Dieser Vorgang benötigt, neben Bandbreite, vor allem Zeit. In dieser Zeit steht die mobile Station somit dem Routing des Multihop-Routing-Protokolls nicht zur Verfügung, so dass daraus eine Beeinträchtigung der Leistungsfähigkeit des WMNs entstehen kann. Indem eine mobile Station die Reichweite ihrer Nachbarstationen verlässt, ergeben sich daraus ebenfalls Konsequenzen für die Nachbarstationen. So muss die mobile Station deauthentifiziert werden, so dass der GTK erneuert werden muss. Die Erneuerung des GTKs benötigt allerdings Bandbreite, die in Abhängigkeit von der Größe des Netzwerks unterschiedlich stark ausfallen kann und so für Nutzdaten nicht zur Verfügung steht.

[1] Zeit zwischen dem Einschalten und der vollständigen Kommunikationsfähigkeit aller Stationen.

Die Optimierung des Datendurchsatzes der Nutzdaten durch eine Vereinfachung des Schlüsselmanagements, sowie der Authentifizierung würde der Leistungsfähigkeit und Flexibilität, im Hinblick auf Anwendungsszenarien, in denen Bandbreitengarantien gegeben werden müssen, wie beispielsweise in der Prozessindustrie, bei der Steuerung von mobilen Robotern oder im Bereich von Voice Over IP *(VoIP)*, Anwendungen zugute kommen. Daher ist es sinnvoll die vorhandenen Security-Mechanismen des IEEE 802.11i Standards an die Bedürfnisse von WMNs anzupassen und so einen speziellen WMN-Modus zu schaffen, der einen besseren Kompromiss zwischen der Leistungsfähigkeit, der Flexibilität und der Security *(Skalierung)*, als der IEEE 802.11i Standard, innerhalb von WMNs erzielt. In den folgenden Abschnitten wird ein Konzept, für ein solches Verfahren, Schrittweise entwickelt.

3.4.2 Authentifizierung

Das in *Abschnitt 3.1* beschriebene Verfahren der paarweisen Authentifizierung des IEEE 802.11i Standards für den Ad-hoc-Modus belastet das Medium mit zunehmender Stationsanzahl quadratisch. Diese Belastung des Mediums kann gravierend reduziert werden, indem eine nicht authentifizierte Station sich lediglich mit einer passenden, bereits authentifizierten Station, authentifiziert. Dies ist in WMNs möglich, da alle zu einem WMN gehörenden Stationen, vertrauenswürdig sein müssen, andernfalls wäre das Routing von Nachrichten, mit Hilfe eines Multihop-Routing-Protokolls, nicht realisierbar.

Wie schon der IEEE 802.11i Standard nutzt der WMN-Modus zur Authentifizierung das 4-Wege-Handshake Protokoll, auf Basis eines PSKs. Dies hat den Vorteil, dass auf die entwickelte Managementsoftware aus *Abschnitt 3.3.4* aufgebaut werden kann. Darüber hinaus stellt der 4-Wege-Handshake ebenso ein robustes und standardisiertes Protokoll zur Authentifizierung dar, so dass aus Sicht der Security an dieser Stelle kein Handlungsbedarf besteht. Da das 4-Wege-Handshake Protokoll außerdem die Authentizität zweier Kommunikationspartner überprüft, wird eine Authentifizierung zwischen zwei Stationen im WMN-Modus lediglich mit der Durchführung eines 4-Wege-Handshake überprüft. Damit ändert sich allerdings auch die aus dem IEEE 802.11i Standard bekannte Rollenverteilung der Stationen. So muss festgelegt werden, welche Station den 4-Wege-Handshake initiiert und damit die Rolle eines Authenticators einnimmt. In Anlehnung an die Definition eines Authenticators *(siehe Abschnitt 2.3.3; 4-Wege-Handshake)* ist ein Authenticator eine Instanz, die anderen Stationen einen Dienst zur Verfügung stellt. Da ein WMN ein Netzwerk mit etwaiger Anbindung an weitere Netzwerke, beispielsweise dem Internet, darstellt und innerhalb eines WMNs sämtliche Stationen frei miteinander kommunizieren können, verfügen so sämtliche Stationen über einen Dienst, den sie anbieten können. Daraus ergibt sich, dass jede authentifizierte Station, innerhalb eines WMNs, die Rolle eines Authenticators inne hat, während jede nicht authentifizierte Station einen potentiellen Supplicant darstellt. Diese Festlegung genügt allerdings noch nicht, da sie den Initialen Zustand eines

WMNs nicht berücksichtigt. Da WMNs aus Stationen bestehen, die im Ad-hoc-Modus betrieben werden, ist jede Station nach der Initialisierung als gleichwertig anzusehen und damit nach der vorangegangenen Festlegung ein Supplicant. Eine Authentifizierung könnte so nicht erfolgen. Daher muss eine Station ausgewählt werden, die die Rolle des Authenticators nach der Initialisierung übernimmt. Hierfür ist eine manuelle Lösung denkbar, indem der Anwender eine Station auswählt, die die Rolle des Authenticators nach der Initialisierung übernehmen soll oder ein automatisiertes Verfahren, welches eine entsprechende Station auswählt. Letzteres kann mit Algorithmen der *Leader Election*, aus den Vorarbeiten [MWV00, HPS99], realisiert werden, so dass anhand von festgelegten Kriterien, wie z.B. der kleinsten MAC-Adresse, eine Station ausgewählt wird. Auch eine Auswahl durch einen Server im Backbone ist denkbar, um den Konfigurationsaufwand noch weiter zu reduzieren. Beide Möglichkeiten werden aber im Hinblick auf die Entwicklung eines einfachen Prototyps nicht weiter verfolgt.

Eine manuelle Auswahl des Initialen Authenticators ist in der Mehrheit der Anwendungsszenarien von WMNs vollkommen ausreichend. So ist eine Station meist ohnehin mit einem weiteren Netzwerk fest, z.B. dem Internet, verbunden oder nimmt anderweitig eine stationäre Position ein. Für eine solche Station kann die Annahme getroffen werden, dass sie sich in einem Vertrauenswürdigen Umfeld befindet, somit leicht zugänglich ist und daher auch einfach die Rolle des Authenticators nach der Initialisierung zugeteilt bekommen kann. Auch ist es so leicht, eine solche Station im Bedarfsfall redundant auszulegen, wodurch kein Verfügbarkeitsproblem entstünde.

Um eine solche Station, die nach der Initialisierung die Rolle des Authenticators einnimmt, leichter von den übrigen Stationen abgrenzen zu können, wird diese fortan als *Master-Station* bezeichnet. Die übrigen Stationen, in Analogie daran, als *Client-Stationen*. Nachdem geklärt wurde, welche Stationen zu welchem Zeitpunkt die Rolle des Authenticators bzw. des Supplicants einnehmen, gilt es noch zu klären wie der Supplicant dem Authenticator mitteilt, dass er authentifiziert werden möchte. Hierfür stehen zwei grundsätzliche Methoden zur Verfügung. Bei dem Konzept des IEEE 802.11i Standards für den Ad-hoc-Modus erfolgt die Initiierung des 4-Wege-Handshakes, durch ein implizit im Treiber erzeugtes *New Station*-Ereignis *(siehe Abschnitt 3.2)*. Diese Möglichkeit kann so im WMN-Modus nicht genutzt werden, da die Erzeugung des Ereignisses lediglich auf Grundlage von empfangenen Beacon-Frames getroffen wird. Allerdings enthält ein Beacon-Frame keine spezielle Information darüber, ob und mit welcher Station eine Authentifizierung durchgeführt wird bzw. durchgeführt werden soll. Damit ist diese Art der Initiierung des 4-Wege-Handshakes weniger praktikabel, als eine zweite bereits im Infrastruktur-Modus verwendete Lösung. Mit der Einführung einer speziellen Station, der Master-Station, existiert in WMNs nunmehr bereits zu Beginn eine zentrale Instanz. Nachdem eine Client-Station sich mit der Master-Station authentifiziert hat, wechselt diese ebenfalls von der Rolle des Supplicants in die Rolle des Authenticators, da sie nunmehr den Dienst des WMNs in vollem Umfang widerspiegelt. Daher lässt sich einfach das Konzept der expliziten Assoziierung, aus dem Infrastruktur-Modus, in den WMN-Modus portieren. Hierbei wählt der Supplicant einen Authenticator aktiv oder passiv

aus, zu dessen Zelle er gehören möchte. Auf den WMN-Modus übertragen bedeutet dies, dass der Supplicant einen Authenticator aktiv oder passiv auswählt, um authentifiziert zu werden. Geschieht dies passiv, so muss jeder Authenticator in regelmäßigen Abständen Nachrichten senden, die angeben, dass ein Authenticator zur Verfügung steht. Im Infrastruktur-Modus wird diese Information bereits in den Beacon-Frames mitgeführt, da der Authenticator per Definition gleichermaßen die Zelle bestimmt. In WMNs wird die Zelle jedoch nicht vom Authenticator festgelegt, sondern ergibt sich implizit durch den Mechanismus des *Beaconings*[1]. Damit müssten zusätzliche Informationen in den Beacon-Frames untergebracht werden, was aber entgegen dem IEEE 802.11 Standard wäre und zu Inkompatibilitäten führen könnte. Als einfacher und darüber hinaus auch Standardkonform wäre die Nutzung von speziellen Daten-Frames. Diese erlauben eine Kapselung von beliebigen Informationen und können ebenfalls in regelmäßigen Abständen als Broadcast-Nachrichten versendet werden. Der Supplicant könnte beim Empfang eines solchen, speziellen, Daten-Frames sich einen Authenticator auswählen und diesen wiederum mit einem eigenen speziellen Daten-Frame mitteilen, dass er authentifiziert werden will. Allerdings besteht ein WMN im Allgemeinen aus einer Vielzahl von Authenticators, so dass ein Versenden solcher speziellen Daten-Frames eine nicht unerhebliche, zusätzliche, Belastung des Mediums darstellen würde, während möglicherweise nur ein Supplicant dem WMN beitreten möchte. Daher wäre eine aktive Auswahl des Supplicants performanter. In diesem Szenario sendet der Supplicant in regelmäßigen Abständen spezielle Broadcast-Nachrichten und stellt damit an andere Stationen die Anfrage einer Authentifizierung. Stationen die dann die Rolle des Authenticators inne haben, könnten dann sofort den 4-Wege-Handshake mit dem Supplicant initiieren. Da es sich allerdings um eine Broadcast-Nachricht handelt und eine Vielzahl von Stationen die Rolle des Authenticators inne haben könnten, würden auch mehrere 4-Wege-Handshakes initiiert werden. Der Supplicant würde dann einfach anhand eines Kriteriums, beispielsweise der zuerst empfangenen Antwort, den Authenticator auswählen und mit der Authentifizierung fortfahren. Dies wird ermöglicht, indem die empfange Antwort des Authenticators bereits mit der ersten Nachricht des 4-Wege-Handshakes übereinstimmt. Die übrigen Antworten werden einfach ignoriert. Zwar belasten die Antworten das Medium ebenfalls, allerdings handelt es sich im Vergleich zur passiven Variante dabei um deutlich weniger Nachrichten. Mit der aktiven Auswahl des Authenticators, mittels einer expliziten Anfrage, ist es nunmehr möglich eine Authentifizierung zwischen einem Supplicant und einem Authenticator, unter der Verwendung nur eines 4-Wege-Handshakes, durchzuführen, so dass sich für vier Stationen das in Abbildung 3.9 dargestellte Bild im WMN-Modus ergibt.

[1] Die Zellenzuordnung erfolgt anhand von Informationen in den Beacon-Frames.

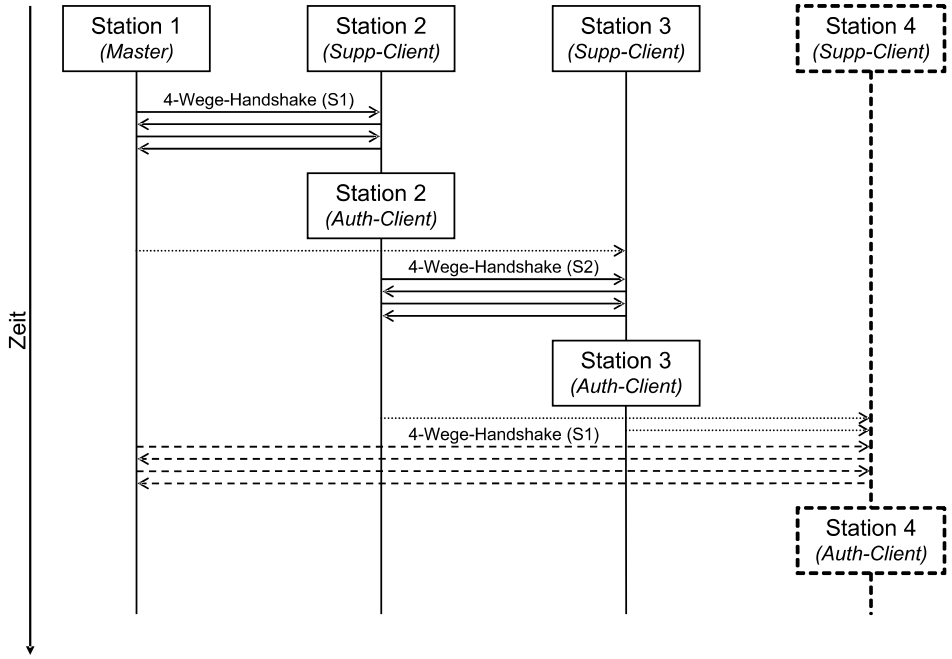

Abbildung 3.9: Authentifizierung in WMNs

Zu Beginn können sich sämtliche Stationen nur bei Station 1 authentifizieren, da sie die Master-Station darstellt. Nachdem Station 2 sich bei Station 1 authentifiziert hat, wechselt diese in die Rolle eines Authenticators, so dass es Station 3 möglich ist eine Authentifizierung mit Station 1 oder Station 2 vorzunehmen. Da Station 3 via Broadcast-Nachrichten mitteilt, dass sie authentifiziert werden will, antwortet sowohl Station 1, als auch Station 2 auf diese Nachricht. Station 2 geht eine Authentifizierung mit Station 1 ein, da sie beispielsweise zeitnäher beantwortet wurde. Nachdem die Authentifizierungen ebenfalls erfolgreich verlief, wechselt Station 3 ebenfalls in die Rolle des Authenticators. Würde nun eine Station 4 hinzukommen, würde diese ebenfalls nur eine Authentifizierung mit einer der drei nun zur Verfügung stehenden Stationen eingehen. Daraus lässt sich folgender allgemeiner Kommunikationsaufwand zur Authentifizierung für *n*-Stationen ableiten.

$$O((n-1) * 1) = O(n * 1) = \underline{\underline{O(n)}}$$

Mit Ausnahme der Master-Station muss jede Station den 4-Wege-Handshake einmal durchführen. Dies bedeutet, dass mit einer zunehmenden Anzahl von Stationen die für die Authentifizierung benötigte Belastung des Mediums proportional ansteigt.

3.4.3 Schlüsselmanagement

Wie bereits in *Abschnitt 2.3.3* aufgeführt stellt ein Schlüsselmanagement einen entscheidenden Punkt zur Gewährleistung von Security dar. Daher sieht der IEEE 802.11i Standard ein Schlüsselmanagement vor, welches ebenfalls in *Abschnitt 3.3.2* für den Ad-hoc-Modus beschrieben wurde. Da der IEEE 802.11i Standard die Vertraulichkeit und Integrität für jede Unicast-Verbindung und Multicast-Verbindung

separat gewährleistet, benötigt jede Verbindung ebenfalls separate Schlüssel. In der Abbildung 3.10 wird dies an einem Beispiel von drei Stationen zusammenfassend dargestellt.

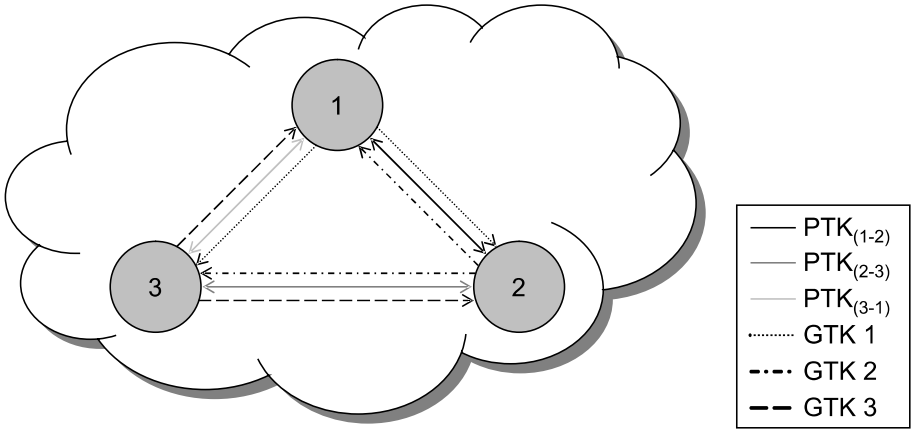

Abbildung 3.10: Benötigte Schlüssel nach dem IEEE 802.11i Standard

In *Abschnitt 3.4.1* wurde erkannt, dass zur Wahrung der Verbindlichkeit und Integrität in einem WMN ein Schlüssel für alle Verbindungen genügt. Da ein Schlüssel, auch nach dem IEEE 802.11i Standard, nicht direkt zur Verschlüsselung genutzt wird, sondern neben dem IV auch die MAC-Adresse in das tatsächlich zur Verschlüsselung verwendete Schlüsselderivat einfließt, entsteht nicht das Problem, dass der gleiche Schlüssel von mehreren Stationen verwendet wird. Zudem erleichtert die Verwendung eines Schlüssels die Schlüsselverwaltung, beispielsweise gegenüber dem IEEE 802.11i Standard. In Abbildung 3.11 ist die so entstandene Situation aus Abbildung 3.10 dargestellt.

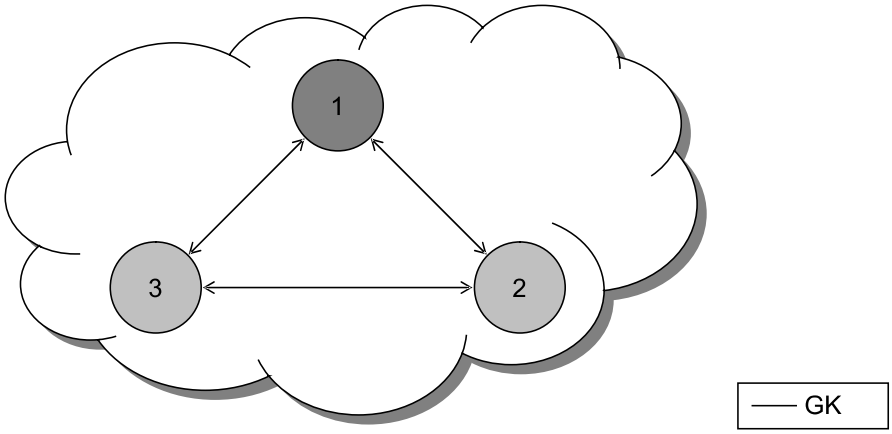

Abbildung 3.11: Benötigte Schlüssel im WMN-Modus

Das gesamte WMN verwendet so nur noch einen Schlüssel, den *Global Key (GK)*, zur Sicherung aller Verbindungen. Damit entfällt eine Unterscheidung zwischen Unicast-Verbindungen und Multicast-Verbindungen innerhalb eines WMNs, was bereits der im vorangegangen Abschnitt behandelten Authentifizierung zugute kam. Allerdings muss der GK, bevor er dem WMN zur Verfügung steht, erst einmal erzeugt werden.

Für die Erzeugung des GKs können zwei grundlegend verschiedene Ansätze verfolgt werden. Der erste Ansatz besteht in einer dezentralen Erzeugung des GKs. Dabei könnten einerseits die Konzepte des *Secret Splittings* oder des *Secret Sharings* [Sch96] einfließen oder andererseits eine Leader Election [MWV00] verwendet werden. Beim Secret Splitting bzw. allgemeiner beim Secret Sharing würde der GK so im WMN verteilt werden, dass jede Station einen Teil des GKs vorhält und der komplette GK sich aus einer gewissen Anzahl von Teilen erzeugen lässt. Dies würde die Angriffsresistenz stärken, allerdings den Aufwand der Schlüsselermittelung und damit möglicherweise den Kommunikationsaufwand erhöhen. Daher verfolgt der Prototyp diesen Ansatz nicht weiter. Die Leader Election könnte dazu genutzt werden eine Station auszuwählen, die für die Erzeugung und später auch für die Erneuerung des GKs verantwortlich ist, ähnlich der im vorangegangenen Abschnitt eingeräumten Möglichkeit der Auswahl eines Initialen Authenticators. Im Hinblick auf die Mehrheit der Anwendungsszenarien von WMNs, in denen eine stationäre Station die Anbindung an ein weiteres Netzwerk darstellt, kann jedoch davon ausgegangen werden, dass sich mindestens eine Station in einem Vertrauenswürdigen Umfeld befindet, die diese Aufgabe übernimmt. Im Bedarfsfall könnte diese auch redundant ausgeführt werden. Somit könnte der Kommunikationsaufwand einer Leader Election entfallen und ein zentraler, manueller Ansatz, wie schon bei der Auswahl eines Initialen Authenticators, zum Einsatz kommen. Dieser Ansatz wird daher auch für die Entwicklung des Prototyps weiter verfolgt.

Da die Auswahl des Initialen Authenticators bereits manuell erfolgt, ist es sinnvoll diese Auswahl gleichermaßen für die Auswahl der für die Erzeugung und Erneuerung des GKs zuständigen Station zu nutzen. Somit ist im WMN-Modus die vom Anwender einmalig festgelegte Master-Station *(Station 1)*, für die Erzeugung und Erneuerung des GKs verantwortlich. Darüber hinaus ergibt sich durch die manuelle Auswahl der Master-Station noch ein weiterer Vorteil, so müssen die Client-Stationen keinen Mechanismus zur Schlüsselerzeugung implementieren, so dass hierfür auch Systeme mit vergleichbar geringeren Ressourcen genügen.

Zur Erzeugung eines kryptographisch starken GKs könnten verschiedene Methoden [Sch96] Verwendung finden. Mit der Umsetzung des IEEE 802.11i Standards steht bereits eine Möglichkeit zur Verfügung, so dass es sinnvoll ist diese auch weiter zu verwenden. So wird der GK, analog zum GTK des IEEE 802.11i Standards, bestimmt und könnte so gleichermaßen, unter Verwendung des Group Key-Handshake Protokolls, verteilt werden. Allerdings würde sich an dieser Stelle ein Konflikt mit dem derzeitigen Stand des Konzeptes ergeben, da das Group Key-Handshake Protokoll zwischen der verteilenden Station des GKs und der Empfängerstation jeweils eine gesicherte Unicast-Verbindung erwartet, diese aber im WMN-Modus nicht länger besteht. Abhilfe an dieser Stelle schafft eine einfache Erweiterung der Authentifizierung. Da bei der Verwendung des 4-Wege-Handshakes zur Authentifizierung weiterhin ein PTK erzeugt wird, wird dieser, noch vor dem Verwerfen, einmalig für die Verteilung des GKs unter Nutzung des Group Key-Handshakes verwendet. Damit erhält jede authentifizierte Station unmittelbar nach ihrer Authentifizierung den GK. Einher geht damit, dass es

weiterhin unerheblich ist, mit welcher authentifizierten Station die Authentifizierung durchgeführt wurde, da jede authentifizierte Station nun gleichermaßen auch den GK kennt. Um dies immer zu garantieren, muss allerdings festgelegt werden, dass der Wechsel von der Rolle des Supplicants zur der des Authenticators nicht sofort nach der Authentifizierung vollzogen wird, sondern erst nach dem Erhalt des GKs. Alternativ zum Group Key-Handshake könnte so auch einfach der GK, bereits während des 4-Wege-Handshakes verteilt werden, was zu einer geringfügigen Reduzierung des Kommunikationsaufwands führen würde.

Aus der nun verwendeten Kombination von Authentifizierung und Schlüsselverteilung folgen weitere Konsequenzen, innerhalb des Schlüsselmanagements. So wird die Schlüsselaktualisierung nunmehr ebenfalls mit einer Re-Authentifizierung kombiniert. Der Zusammenhang zwischen der Re-Authentifizierung und der Schlüsselaktualisierung, sowie die hinter der Schlüsselaktualisierung stehenden Mechanismen werden nachfolgend, in einem gesonderten Abschnitt, ausgiebig betrachtet.

3.4.4 Re-Authentifizierung und Schlüsselaktualisierung

Mit dem Verwerfen des PTKs, nach der Übermittlung des GKs, entfällt die Möglichkeit den PTK für eine Verteilung, nach der Erneuerung des GKs, zu nutzen. Auch wenn der PTK gesichert werden würde, so könnte in einem WMN, aufgrund von Mobilität, nicht garantiert werden, dass die Unicast-Verbindung zum Zeitpunkt der Schlüsselaktualisierung noch besteht. Daher wird festgelegt, dass eine Schlüsselaktualisierung mit einer Re-Authentifizierung verknüpft wird. So bleibt der erhöhte Schutz der Verteilung des GKs, durch die einmalige Verwendung des PTKs zur Verschlüsselung, erhalten. Durch das erneute aushandeln des PTKs stellt damit die Mobilität von Stationen an dieser Stelle ebenfalls kein Problem dar. Auch könnten so einfach weitere Informationen, wie beispielsweise Blacklisten zum Sperren einiger Stationen, gesichert, verteilt werden und der Security kommt eine erneute Überprüfung der Authentizität *(siehe Abschnitt 2.3.1)* ohnehin zugute.

Wie bereits im vorangegangenen *Abschnitt 3.4.3* beschrieben, ist für die Aktualisierung des GKs die Master-Station zuständig. Somit bestimmt sie ebenfalls den Zeitpunkt der Schlüsselaktualisierung. Da mit der Schlüsselaktualisierung eine Re-Authentifizierung kombiniert werden soll, muss den Client-Stationen mitgeteilt werden, wann eine Re-Authentifizierung ansteht. Dies kann auf zwei verschiedene Arten realisiert werden. Entweder wird jeder Client-Station bereits bei der Authentifizierung mitgeteilt, wann der Zeitpunkt der erneuten Authentifizierung ansteht oder der Zeitpunkt wird, durch die Master-Station ausgelöst, durch das gesamte WMN propagiert. Letzteres hätte den Vorteil, dass während der Authentifizierung keine zusätzlichen Informationen über die zeitliche Gültigkeit des Schlüssels mitgeteilt und von den Client-Stationen verwaltet werden müssten. Allerdings kann es aufgrund von Mobilität und damit verbunden, einem möglichen temporären Verlust der Konnektivität, dazu kommen, dass einzelne

Stationen die Nachricht der Schlüsselaktualisierung in Kombination mit einer Re-Authentifizierung nicht erhalten. Ist dies der Fall, so würden die entsprechenden Stationen keinen neuen Schlüssel erhalten und bis zu einem manuellen zurücksetzen ein eigenes WMN bilden. Daher baut der WMN-Modus darauf dem GK eine zeitliche Gültigkeit bei der Schlüsselverteilung zu zuordnen. Damit ist jede Client-Station, analog zur Initialen Authentifizierung, selbst für die Re-Authentifizierung verantwortlich. Kann sich eine Client-Station bis zum Ablauf der zeitlichen Gültigkeit des GKs nicht erneut Authentifizierung, so wechselt sie zurück in ihre Initiale Rolle des Supplicants und muss sich erneut Authentifizieren. In Abbildung 3.12 wird der bisher im Rahmen des Konzepts geschilderte Ablauf zwischen Authentifizierung und Re-Authentifizierung in einem Zustandsdiagramm zusammenfassend dargestellt.

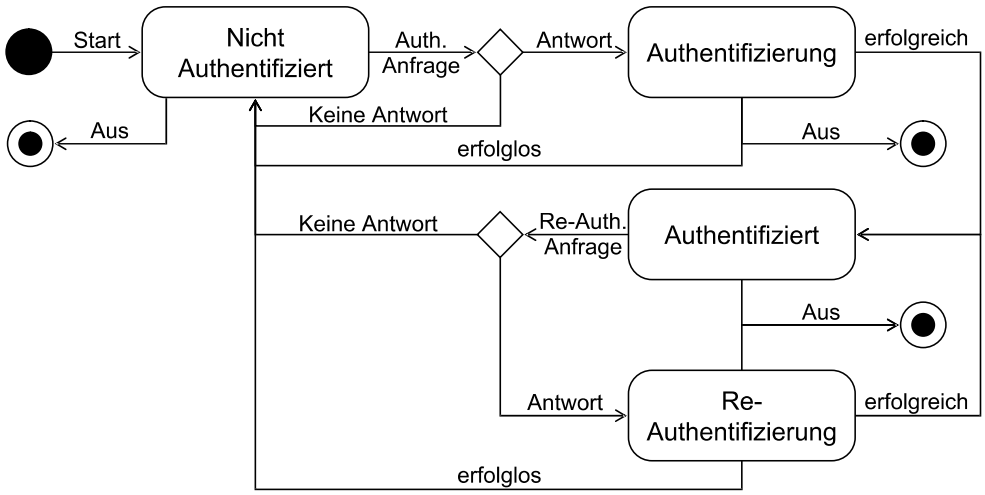

Abbildung 3.12: Zustandsdiagramm der Authentifizierung einer Client-Station

Um auf eine Uhrensynchronisation zwischen den Stationen verzichten zu können, wird bei der Verteilung des GKs durch den Authenticator, neben dem GK auch die zugehörige relative zeitliche Gültigkeit übermittelt. Jede Station kann so den Zeitpunkt der Schlüsselaktualisierung bestimmen und in Kombination mit einer Re-Authentifizierung eine Aktualisierung des GKs vornehmen. Allerdings entsteht mit der Aktualisierung des GKs auch ein Problem. So ist es unmöglich, dass alle Stationen zum gleichen Zeitpunkt eine Re-Authentifizierung durchführen und darüber hinaus auch noch zeitgleich den GK aktualisieren können. Es würde eine Situation entstehen in der einige Stationen bereits den aktualisierten GK verwenden, während andere Stationen noch mit dem alten GK kommunizieren. Die Folge wäre ein vorübergehender Verlust der Konnektivität zwischen einigen Stationen, so dass es zu Paketverlusten während einer etwaigen Datenübertragung kommen würde.

Zur Begegnung dieses Problems, wird die Schlüsselaktualisierung im WMN-Modus innerhalb einer Übergangsphase durchgeführt. Während dieser Übergangsphase müssen zwei Schlüssel temporär verwaltet werden können. Dies stellt jedoch kein Problem dar, da der IEEE 802.11i Standard eine Schlüsselverwaltung von mindestens vier Schlüssel in einer Indexierbaren Schlüsseltabelle vorsieht. Mit einer zeitlichen Differenz *2d*, vor Überschreitung der zeitlichen Gültigkeit des GKs beginnt diese Übergangsphase.

Zunächst wird die Re-Authentifizierung eingeleitet und damit einher auch der neue GK angefordert. Zur Einleitung der Re-Authentifizierung, versendet die Client-Station, aktiv, in regelmäßigen Abständen spezielle Broadcast-Nachrichten. Diese Broadcast-Nachrichen fungieren als Anfrage für die Re-Authentifizierung an bereits Re-Authentifizierte Stationen. An dieser Stelle ist eine Unterscheidung zwischen den Broadcast-Nachrichten der Authentifizierung und der, der Re-Authentifizierung erforderlich, da eine explizite Anforderung eines neuen GKs damit verbunden ist.

Nachdem der neue GK bezogen wurde, wird er zunächst für eine Zeit *d* nicht aktiv zur Verschlüsselung von Nachrichten verwendet. So wird eine Übergangszeit geschaffen, in der es anderen Client-Stationen weiterhin möglich ist, mit der betrachteten Station zu kommunizieren und ihrerseits eine Re-Authentifizierung durchzuführen. Nachdem Verstreichen der Zeit *d* kann davon ausgegangen werden, dass alle Client-Stationen mit Konnektivität zur betrachteten Station die Re-Authentifizierung durchgeführt haben und damit im Besitz des neuen GKs sind, so dass dieser schließlich zur Verschlüsselung verwendet werden kann. Der alte GK wird ab diesem Zeitpunkt noch bei Bedarf zum Entschlüsseln verwendet, da die Client-Stationen mit Konnektivität zur betrachteten Station ihrerseits noch den alten GK zur Verschlüsselung verwenden. Nach einem weiteren Verstreichen einer Zeitspanne *d* ist schließlich der alte GK auch nicht länger als Empfangsschlüssel gültig, womit die Schlüsselgültigkeit beendet ist. Ab diesem Zeitpunkt gilt nur noch der neue GK zum Verschlüsseln, sowie zum Entschlüsseln von Nachrichten. Damit ist die Schlüsselaktualisierung abgeschlossen. Diese Übergangsphase ist in der Abbildung 3.13, separat für zwei Stationen (t_{A1} und t_{A2}), dargestellt, wobei der aktiv zur Verschlüsselung genutzte GK fett hervorgehoben ist.

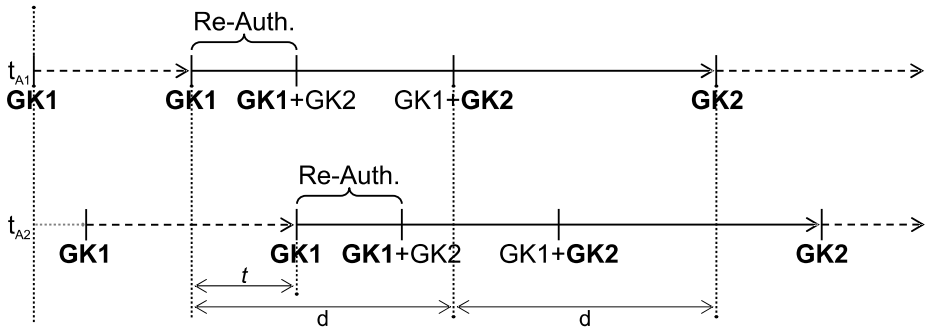

Abbildung 3.13: Übergangsphase zur Schlüsselaktualisierung

Mit der Einführung einer Re-Authentifizierung, in Kombination mit einer Schlüsselaktualisierung, ändert sich ebenfalls die Rollenverteilung, die eine Client-Station ausübt, nahezu periodisch über die Zeit betrachtet. In Abbildung 3.14 sind die Rollen, die zugehörige Aufgabe, die verwendeten Schlüssel und deren Verwendungszweck für eine Client-Station in einem zeitlichen Verlauf exemplarisch dargestellt.

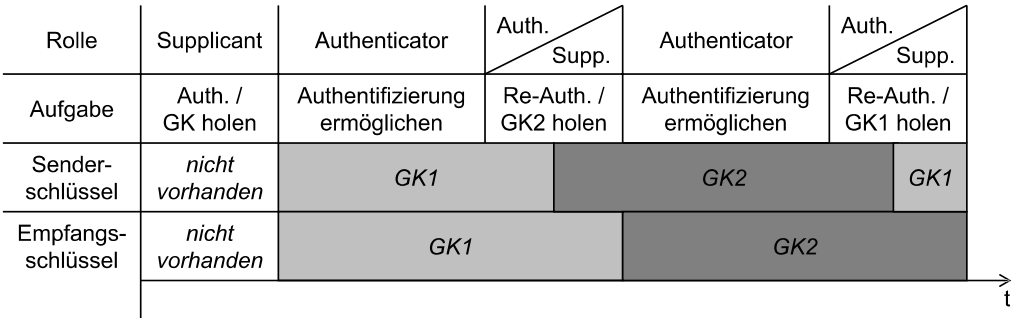

Abbildung 3.14: Rollenverteilung einer Client-Station

Nach der Initialisierung hat eine Client-Station die Aufgabe sich zu authentifizieren. Ist dies erfolgreich geschehen, so kann die Client-Station Nachrichten Ver- und Entschlüsseln. Gleichzeitig nimmt sie nun die Rolle eines Authenticators ein und ermöglicht so, anderen Client-Stationen, eine Authentifizierung. Mit dem Erreichen des Zeitpunktes der Re-Authentifizierung nimmt die Client-Station die Rolle eines Supplicants ein, um eine Verteilung, eines zeitlich nah verbrauchten Schlüssels, an andere Client-Stationen zu unterbinden. Da der GK zu diesem Zeitpunkt weiterhin noch gültig ist, behält die Client-Station trotzdem den Status einer authentifizierten Client-Station bei. Nach der erfolgreichen Re-Authentifizierung nimmt die Client-Station wieder die Rolle des Authenticators ein und ermöglicht so anderen Client-Stationen eine Authentifizierung, sowie eine Re-Authentifizierung. Für eine Übergangszeit *d* wird der neue GK nur als Empfangsschlüssel verwendet, so dass der neue GK zunächst erst einmal verteilt werden kann. Anschließend wird der alte GK, solange er noch gültig ist, als Empfangsschlüssel verwendet. Nach Ablauf der Gültigkeit ist die Schlüsselaktualisierung schließlich abgeschlossen und der Ablauf beginnt nach einer gewissen Zeit von vorn.

3.4.5 Mechanismus gegen Replay-Angriffe

Der IEEE 802.11i Standard nutzt zur Verhinderung von Replay-Angriffen einen Sequenzzähler, den jede Station lokal vorhält. Dieser wird dazu verwendet eine logische Reihenfolge zwischen den Daten-Frames herzustellen, indem er für jedes Daten-Frame inkrementiert wird, so dass noch nicht empfangene Daten-Frames einen größeren Wert, gegenüber dem lokalen Wert, aufweisen. Damit wird ein bereits empfangenes Daten-Frame kein zweites Mal akzeptiert. Der IEEE 802.11i Standard nutzt für den Sequenzzähler den IV, da dieser mit einem Schlüssel ohnehin nur einmal verwendet werden sollte kann er bei Null initialisiert werden und im Verlauf inkrementiert werden. Damit existiert ein für jeden Schlüssel einzigartiger Sequenzzähler, der bei einer Schlüsselaktualisierung ebenfalls zurückgesetzt wird und damit eine logische Reihenfolge zwischen den Daten-Frames einer Kommunikationsbeziehung beschreibt.

Nachdem der WMN-Modus nunmehr sämtliche Nachrichten, unter Verwendung nur eines GKs verschlüsselt, hat dies ebenfalls Konsequenzen für den Mechanismus gegen Replay-Angriffe. So ist der Mechanismus des IEEE 802.11i Standards, bei dem der IV

in Kombination mit dem zugehörigen Schlüssel, gleichzeitig eine logische Ordnung widerspiegelt, nicht verwendbar. Die Ursache hierfür liegt in der Verwendung des GKs zur Verschlüsselung und Entschlüsselung von sämtlichen Stationen, innerhalb des WMNs. So ist es denkbar, dass zwei Stationen miteinander kommunizieren, ebenso wie zwei weitere, innerhalb der Reichweite des ersten Stationspaares. Damit würden die Zähler zunächst für den Schlüssel, in Kombination mit jedem der beiden Stationspaare, übereinstimmen. Will jetzt allerdings, beispielsweise eine Station des ersten Stationspaares mit einer des zweiten kommunizieren, so ist einer von beiden Zählern niedriger. Sendet die Station mit dem niedrigeren Zähler, der anderen Station, eine Nachricht, so würde diese als logisch älter angesehen werden und damit ignoriert werden. Damit greift dieser Mechanismus für den WMN-Modus nicht.

Eine einfache Form, zumindest die Wahrscheinlichkeit von erfolgreichen Replay-Angriffen zu minimieren, besteht in einem häufigen aktualisieren des GKs. Damit würde sich der Zeitraum, in dem ein Replay-Angriff erfolgreich durchgeführt werden kann, verkürzen. Allerdings sind damit im Wesentlichen zwei Nachteile verbunden. Der erste und große Nachteil ist, dass Replay-Angriffe so nicht gänzlich ausgeschlossen werden können. Ein zweiter Nachteil resultiert aus dem Mechanismus der Schlüsselaktualisierung. So ist mit jeder Aktualisierung des GKs, ebenfalls eine Re-Authentifizierung verbunden. Diese benötigt Bandbreite auf dem Medium, sowie Ressourcen zum Versenden und Empfangen von Nachrichten des WLAN-Gerätes, so dass Anwendungen eine geringere Bandbreite zur Verfügung gestellt wird.

Um dennoch, auch im WMN-Modus einen Mechanismus gegen Replay-Angriffe anbieten zu können, kann die aus dem IEEE 802.11i Standard bekannte Überprüfung der logischen Ordnung in modifizierter Form eingesetzt werden. So könnte jede Station einen logischen Zähler mitführen, der nicht an einen Schlüssel gekoppelt, sondern an die MAC-Adresse der Absenderstation gekoppelt wird. Damit könnten Nachrichten die einen kleineren Zählerwert aufweisen, also demnach schon einmal in der Vergangenheit versendet worden sind, ignoriert werden und Replay-Angriffe wären nicht möglich. Allerdings hat dieses Konzept ein Problem, wenn eine Station ausfällt und anschließend neu gestartet wird. In diesem Fall würde der lokale Zählerwert wieder bei Null initialisiert werden, so dass die versendete Nachricht aus Sicht der Empfängerstation als logisch älter und damit als ungültig einzustufen wäre. Erst eine spätere Schlüsselaktualisierung würde daran etwas ändern. Unter der Einschränkung, dass sich die Nachbarstationen, der ausgefallenen Station nicht geändert haben, wäre ebenso eine Übertragung, des höchsten Wertes der Zähler der Nachbarstationen, zur Initialisierung des lokalen logischen Zählers, während der Authentifizierung denkbar. Allerdings würde dies eine starke Einschränkung des WMNs bedeuten, so dass die Annahme getroffen wird, dass ein solcher Fall selten auftritt. Tritt ein solcher Fall auf, so ist die Station nach dem Neustart bis zum nächsten Zeitpunkt der Schlüsselaktualisierung nicht nutzbar. Eine Alternative würde das frühzeitige Erkennen eines Ausfalls und eine daraufhin eingeleitete Schlüsselaktualisierung darstellen. Da das Erkennen eines Ausfalls jedoch nicht ohne weiteres realisierbar ist, wird an dieser Stelle darauf verzichtet. Somit steht es dem Anwender frei, auf Kosten von Bandbreite für die

Nutzdaten, die Zeitperiode der Schlüsselaktualisierung zu verkürzen und so im Falle eines Neustarts, einer bereits in der Vergangenheit authentifizierten Station, deren Zeit zur vollen Kommunikationfähigkeit zu verkürzen. Im Normalbetrieb stellt der Neustart einer Station eine Ausnahme dar, so dass der stationsgebundene logische Zähler einen effizienten Schutz gegen Replay-Angriffe im WMN-Modus darstellt.

3.4.6 Softwaredesign

Der WMN-Modus benötigt lediglich einen Treiber mit Unterstützung des IEEE 802.11i Standards im Infrastruktur-Modus. Alle notwendigen konzeptionellen Änderungen können in einer Managementsoftware realisiert werden, so dass eine Vielzahl von WLAN-Geräten unterstützt werden kann. Wie auch die Managementsoftware für den Ad-hoc-Modus, aus *Abschnitt 3.3.4*, baut der WMN-Modus auf Mechanismen des IEEE 802.11i Standards. Daher werden die entwickelten Mechanismen für WMNs lediglich als weitere optionale Konfigurationsmöglichkeit, in das bestehende Konzept aus Abbildung 3.7 übernommen, so dass sich konzeptionell an dieser Stelle keine Änderungen ergeben.

3.5 Vergleich der entwickelten Verfahren

Der entwickelte WMN-Modus sieht eine Überprüfung der Authentizität einer nicht authentifizierten Station gegenüber dem WMN, die Sicherung der Integrität, sowie der Verbindlichkeit von Nachrichten innerhalb des selbigen vor. Durch die Nutzung des 4-Wege-Handshake-Protokolls, analog zum IEEE 802.11i Standard wird eine gegenseitige Authentifizierung zwischen einer Station aus dem WMN und der nicht authentifizierten Station gewährleistet. Wie in *Abschnitt 2.3.1* beschrieben, ist dies eines von vier Kriterien für eine robuste Authentifizierungsmethode. Weiterhin wird ein eigenständiger Schlüssel zur Authentifizierung verwendet und die Authentizität nach einer gewissen Zeit, durch eine Re-Authentifizierung erneut überprüft. Da der GK für eine Kommunikation mit dem WMN erforderlich ist und dieser, ähnlich zum IEEE 802.11i Standard, erst durch eine erfolgreiche Authentifizierung bekannt gemacht wird, stellt dies keine Abwertung der Security gegenüber dem IEEE 802.11i Standard dar. Die Stärke des Verfahrens zur Authentifizierung hängt somit sowohl beim WMN-Modus, als auch beim IEEE 802.11i Standard von der Implementierung ab. Bei der Verwendung einer kryptographisch starken Zufallsfunktion und einem starken PSK ist derzeit keine Möglichkeit bekannt, die Authentifizierung zu umgehen.

Die Verbindlichkeit und Integrität der Nachrichten wird, analog zum IEEE 802.11i Standard durch das CCMP-Verschlüsselungsprotokoll gewährleistet. Auch in CCMP wurden bisher keine Designschwächen gefunden, die sich für Angriffe ausnutzen lassen könnten. Allerdings nutzt der WMN-Modus, im Gegensatz zum IEEE 802.11i Standard, für sämtliche Verbindungen nur einen Schlüssel. Dies hat zur Folge, dass dieser

Schlüssel häufig verwendet wird, was ein mögliches Security-Risiko darstellen kann. Um dieses Risiko zu reduzieren kann der Anwender die zeitliche Gültigkeit des GK verkürzen, was im Vergleich zum IEEE 802.11i Standard eine häufigere Schlüsselaktualisierung und damit eine etwaige höhere Belastung des Mediums zur Folge hat.

Auch gegen Replay-Angriffe besitzt der WMN-Modus einen Mechanismus. Allerdings hat dieser, je nach seiner Umsetzung, Auswirkungen auf die Performance des WMNs bzw. auf die Zeit, bis zur vollen Kommunikationsfähigkeit einer etwaigen neu gestarteten Station. Der IEEE 802.11i Standard bietet mit der Kopplung von IV und Schlüssel hierfür eine einfache und zugleich performante Lösung an.

Da der WMN-Modus in der vorgestellten Form, im Gegensatz zum IEEE 802.11i Standard, mit einer zentralen Instanz arbeitet, kann dies in einigen Szenarien eine Schwäche darstellen. So könnte die Master-Station ausfallen, so dass das gesamte WMN sich nicht Re-Authentifizieren kann und in Folge dessen die Konnektivität innerhalb des WMN verloren geht. Daher sollte die Master-Station eine möglichst stationäre Position, in einem vertrauenswürdigen Umfeld, einnehmen. Um diesen Single Point of Failure im Hinblick auf die Zuverlässigkeit des WMNs zu verbessern, könnte er zudem redundant ausgeführt werden. Eine weitere Möglichkeit besteht in einer dynamischen Auswahl der Master-Station, so dass bei jeder Re-Authentifizierung die Master-Station neu ausgewählt werden könnte. Zur Auswahl der Master-Station könnten Algorithmen der Leader Election aus den Vorarbeiten [MWV00, HPS99] verwendet werden.

4 Implementierung

Dieses Kapitel beschreibt die Umsetzung der im vorherigen Kapitel entwickelten Konzepte. Zunächst wird hierzu auf die Umsetzung des IEEE 802.11i Standards für den Ad-hoc-Modus eingegangen. Dazu werden die benötigten Treiberspezifischen Anpassungen vorgestellt, bevor auf die Umsetzung der Managementsoftware, auf Basis der für den Infrastruktur-Modus verfügbaren Applikationen *hostapd* und *wpa_supplicant* eingegangen wird. Auf Grundlage dieser Managementsoftware werden schließlich die, für den WMN-Modus benötigten, implementierungsspezifischen Details vorgestellt.

4.1 MadWifi

Wie Eingangs in *Abschnitt 2.4* vorgestellt, handelt es sich bei MadWifi um einen Treiber für WLAN-Geräte. Dieser stellte jedoch nicht die für eine Umsetzung des IEEE 802.11i Standards im Ad-hoc-Modus benötigten Funktionalitäten, im vollen Umfang, zur Verfügung, so dass nachfolgend auf die vorgenommenen Erweiterungen eingegangen wird.

4.1.1 Ereigniserzeugung

Mit dem Fehlen einer zentralen Instanz im Ad-hoc-Modus ist eine explizite Zuordnung einer Station zu einer bestimmten Zelle, wie es im Infrastruktur-Modus der Fall ist, nicht möglich. Daher muss ein anderer Weg gefunden werden der Managementsoftware über ein *New Station*-Ereignis bzw. *Station Leave*-Ereignis mitzuteilen, dass eine Authentifizierung bzw. eine Deauthentifizierung durchzuführen ist. Wie in *Abschnitt 3.2* dargelegt, erfolgt die Ereigniserzeugung daher implizit im Treiber. Der verwendete Treiber MadWifi generiert für den Ad-hoc-Modus lediglich ein *Station Leave*-Ereignis für das Szenario, dass eine Station die Reichweite der betrachteten Station verlässt, so dass für die übrigen Szenarien aus dem Konzept in *Abschnitt 3.2* eine Ereigniserzeugung in MadWifi ergänzt werden muss.

Um eine implizite Zuordnung zwischen den Stationen zu ermöglichen, muss eine Station die in ihrer Nachbarschaft befindlichen Stationen kennen. Die hierfür notwendigen Informationen werden in MadWifi mit Hilfe einer verketteten Liste, die als Nachbarschaftstabelle fungiert, verwaltet. Ob eine Station zur Zelle der betrachteten Station letztendlich gehört oder ob die betrachtete Station zu einer anderen Zelle wechselt wird mit der Funktion *ieee80211_ibss_merge*, in MadWifi, überprüft. Kommt es zu einem Wechsel der Zelle, so wird die Nachbarschaftstabelle zunächst auf inkompatible Stationen überprüft. Befinden sich beispielsweise Stationen in der Nachbarschaft, die ausschließlich den IEEE 802.11b Standard unterstützen, während die betrachtete Station nur im IEEE 802.11g Standard kommuniziert, so wird dies in der Nachbarschaftstabelle vermerkt. Um festzustellen zu können, welche Stationen in der

Vergangenheit und aktuell zur Zelle, der betrachteten Station, gehört haben bzw. gehören, wird in der Nachbarschaftstabelle ein zusätzlicher boolescher Wert *(AssocID)* für jede Station mitgeführt. Nach dem Wechsel zu einer anderen Zelle kann so für sämtliche assoziierten Stationen *(AssocID = True)* ein *Station Leave*-Ereignis erzeugt und vermerkt werden. In analoger Form wird für jede nicht assoziierte *(AssocID = False)* und kompatible Station ein *New Station*-Ereignis erzeugt und vermerkt.

Die Erzeugung der Ereignisse für die verbleibenden zwei Szenarien, dass sich eine andere Station mit der betrachteten Station verbindet und das eine andere Station in eine andere Zelle wechselt, als die in der sich die betrachtete Station befindet, wird über die direkte Auswertung der *Beacon*-Frames realisiert. In Abbildung 4.1 ist das Flussdiagramm dargestellt, welches das Vorgehen bei der Erzeugung der beiden Ereignisse beschreibt.

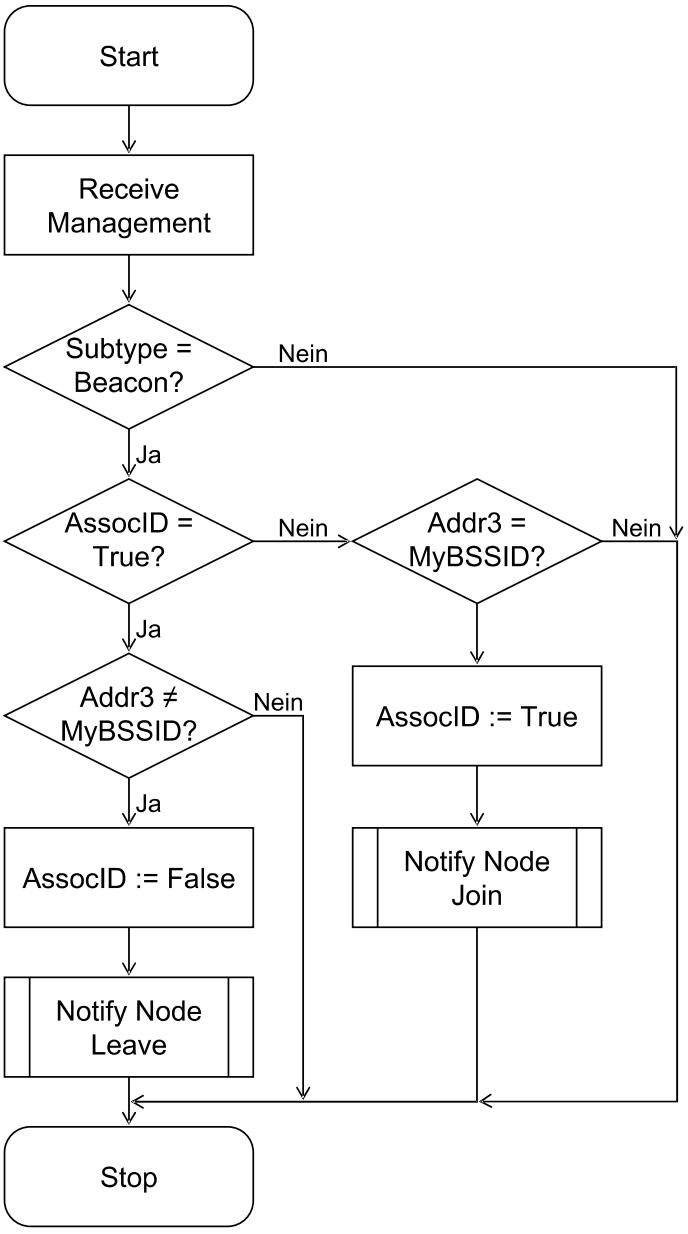

Abbildung 4.1: Flussdiagramm – Direkte Ereigniserzeugung

Nachdem ein *Management*-Frame vom Typ *Beacon* oder *Probe response* empfangen wurde, wird überprüft ob die Station bereits assoziiert *(AssocID = True)* ist oder nicht. Durch eine Überprüfung der Zellen-Adresse der Absenderstation, innerhalb des Frames *(Adressfeld 3)* und der Zellen-Adresse der betrachteten Station *(BSSID)* kann eine Entscheidung bezüglich der Absenderstation getroffen werden. Stimmen die Zellen-Adressen überein, so gehört die Absenderstation zur Zelle der betrachteten Station, andernfalls nicht. Damit kann ein Kontext zur Assoziierungsinformation *(AssocID)* in der Nachbarschaftstabelle hergestellt werden. Ist die Absenderstation mit der betrachteten Station assoziiert *(AssocID = True)* und gehört einer anderen Zelle an, so hat sie einen Zellenwechsel durchgeführt und ein *Station Leave*-Ereignis *(Notify Node Leave)* wird erzeugt und vermerkt. Gehört die Absenderstation zur gleichen Zelle und ist nicht assoziiert *(AssocID = False)*, so hat die Station sich mit der Zelle verbunden und ein *New Station*-Ereignis *(Notify Node Join)* wird erzeugt und vermerkt.

4.1.2 Schlüsselverwaltung und Nutzung

Um die aufwendigen Verschlüsselungsprotokolle auch in Systemen mit vergleichsweise geringen Ressourcen realisieren zu können, werden Teile der Verschlüsselungsprotokolle in speziellen Hardware-Komponenten der WLAN-Geräte realisiert. So wird die Fragmentierung der PTKs und GTKs, sowie die Ver- und Entschlüsselung in der Hardware vorgenommen, so dass lediglich die Schlüsselverwaltung in MadWifi realisiert werden muss. Für die Verwaltung der Schlüssel sieht MadWifi zunächst, analog zu WEP, eine Schlüsseltabelle vor. Dabei kann diese Schlüsseltabelle sowohl PTKs für Unicast-Verbindungen, als auch GTKs für Multicast-Verbindungen verwalten.

Der für die Verschlüsselung verwendete Schlüssel, wird durch die Absender-Station in das Daten-Frame als Schlüsselindex eingetragen, so dass die Empfänger-Station, durch Auslesen des Schlüsselindex, den für die Entschlüsselung benötigten Schlüssel aus der Schlüsseltabelle zuordnen kann. Allerdings ist die Länge eines Schlüsselindex in einem Frame auf zwei Bit festgelegt, so dass maximal vier Schlüssel indiziert werden können.

Auf den Infrastruktur-Modus und einen Authenticator bezogen bedeutet dies, dass eine Kommunikation mit drei Supplicants möglich ist. So wird für jeden Supplicant ein PTK zur Sicherung der Unicast-Verbindung benötigt, sowie ein GTK zur Sicherung von Multicast-Verbindungen. Um mehr als drei PTKs verwalten und nutzen zu können, wird ein zusätzlicher Schlüsselcache eingeführt. Dieser fungiert als eine Art Lookup-Tabelle, indem zu einer MAC-Adresse ein korrespondierender Schlüssel zugeordnet wird. In MadWifi ist ebenfalls ein Schlüsselcache vorgesehen, der Platz für bis zu 128 *(ATH_KEYMAX)* Einträge bietet. Somit kann der Schlüsselcache zur Verwaltung der PTKs genutzt werden, während die Schlüsseltabelle für den GTK verwendet wird.

Für den Ad-hoc-Modus steht dieser Schlüsselcache ebenfalls zur Verfügung, so dass er ebenfalls zur Verwaltung der PTKs verwendet wird. Allerdings benötigt der Ad-hoc-

Modus, wie in Tabelle 3.1 in *Abschnitt 3.1* aufgeschlüsselt ist, außer einem GTK als Versenderschlüssel auch für jede, in Reichweite befindlichen, Station der Zelle jeweils einen GTK als Empfangsschlüssel. Damit könnte jede Station, bei analoger Verwendung zum Infrastruktur-Modus, Multicast-Nachrichten von lediglich drei Nachbarstationen entschlüsseln, obwohl der Schlüsselcache noch Platz für weitere Schlüssel besäße.

Mit der Funktionsweise des Schlüsselcaches, in der genau ein Schlüssel mit genau einer MAC-Adresse korrespondiert, lässt sich dennoch, neben der Verwaltung der PTKs, zusätzlich GTKs als Empfängerschlüssel verwalten. Da ein Broadcast-Frame, die Empfängeradresse *FF:FF:FF:FF:FF:FF* in kanonischer Darstellung aufweist, kann innerhalb von MadWifi eine Neuzuordnung der Absender MAC-Adresse vorgenommen werden. Statt die Absender MAC-Adresse direkt zu verwenden, wird das Broadcast-Bit der Absender MAC-Adresse in kanonischer Darstellung vor der Verwendung gesetzt. Dies erlaubt eine Unterscheidung zwischen dem PTK und dem GTK als Empfangsschlüssel innerhalb des Schlüsselcaches. Die Schlüsseltabelle wird somit nur noch für den GTK als Senderschlüssel benötigt.

4.2 Managementsoftware für den IEEE 802.11i Standard

Nachdem im vorangegangen Abschnitt die Treiberspezifischen Anpassungen des Konzeptes dargelegt wurden, beschäftigt sich dieser Abschnitt mit den Gesichtspunkten der Realisierung einer Managementsoftware im User Space, auf Basis der verfügbaren Applikationen *hostapd (Authenticator)* und *wpa_supplicant (Supplicant)*.

Da viele benötigte Komponenten, beispielsweise die Verwaltung von Gegenstellen, bereits im *hostapd* Verwendung finden, wird die Applikation *hostapd* vollständig in die Managementsoftware integriert, während aus dem *wpa_supplicant* im Wesentlichen die Zustandsmaschinen und deren Peripherie übernommen werden. Daher ist es möglich die Managementsoftware auch weiterhin als Authenticator im *Infrastruktur-Modus* zu betreiben. Nachfolgend wird zunächst auf die Umsetzung einer Konfigurationsschnittstelle eingegangen, bevor ausgehend vom *Treiber Interface* verschiedene Schlüsselstellen, im Hinblick auf die Realisierung einer Managementsoftware, gemäß dem Konzept aus *Abschnitt 3.3.4*, betrachtet werden.

4.2.1 Konfigurationsschnittstelle

Um die Managementsoftware leicht konfigurierbar zu gestalten, setzt die Managementsoftware auf zwei Schnittstellen, analog zum *hostapd*, mit denen die Software konfiguriert wird. Eine Schnittstelle bildet die Verarbeitung einer Konfigurationsdatei. Diese enthält sämtliche Security-Einstellungen, z.B. den PSK, wie auch treiberspezifische Einstellungen, beispielsweise den verwendeten Kanal. So ist es möglich selbige Einstellungen einmalig vorzunehmen und bei Bedarf, die Datei, auch

für andere Systeme zu übernehmen. Der Aufbau der Konfigurationsdatei ist der, des *hostapd* gleichzusetzen, so dass eine Konfiguration weiterhin mit einem Texteditor möglich ist, wie auch die Verwendung des für den *hostapd* und *wpa_supplicant* verfügbaren *GUI Frontends*. Einfache, bereits zur Initialisierung benötigte Parameter, wie z.B. der Pfad zur Konfigurationsdatei, werden hingegen der Software über Programmaufrufparameter mitgeteilt.

Zur Auswertung der Konfigurationsdatei implementiert die Managementsoftware die Komponente *config (config.c)* aus dem *hostapd*. Diese implementiert einen Textparser *(hostapd_config_read)*, der die Konfigurationsdatei Zeilenweise nach Schlüsselwörter und korrespondierenden Werten durchsucht. Wird ein Schlüsselwort gefunden, so werden die zugehörigen Werte in eine interne Datenstruktur *(hostapd_config)* übernommen. Die so erzeugte Datenstruktur *(hostapd_config)* wird schließlich zur Initialisierung der intern benötigten Datenstrukturen verwendet.

Die Auswertung der Programmaufrufparameter benötigt hingegen keine eigenständige Komponente. So ermöglicht, die in der GNU C Bibliothek enthaltende Funktion *getopt* eine bequeme Auswertung von Parametern, so dass *getopt* hierfür Verwendung findet. Neben den aus dem *hostapd* bekannten Aufrufparametern, wird der Managementsoftware ein weiterer Aufrufparameter *-A* hinzugefügt. Dieser gibt an, dass die Managementsoftware im Ad-hoc-Modus betrieben werden soll. Wird dieser Parameter nicht angegeben, so arbeitet die Managementsoftware als Authenticator im Infrastruktur-Modus. Die aus dem *hostapd* übernommene Funktionalität und deren Parameter, sowie die Unterscheidung zwischen Ad-hoc-Modus und Infrastruktur-Modus ergeben, die von der Managementsoftware unterstützten Aufrufparameter. Diese werden, zusammen mit ihrer Funktionalität, in der Tabelle 4.1 zusammenfassend aufgeschlüsselt.

Parameter	**Funktionalität**
-B	Software als Daemon nutzen
-d	Debug-Informationen ausgeben
-h	Hilfe ausgeben
-K	Schlüsselinformationen ausgeben
-P	Prozess-ID in Datei schreiben
-t	Zeitstempel ausgeben
-v	Version der Software ausgeben
-A	Software im Ad-hoc-Modus betreiben

Tabelle 4.1: Konfigurationsparameter der Managementsoftware

4.2.2 Treiberanbindung

Um unabhängig vom verwendeten Treiber, eine Kommunikation zwischen dem Treiber[1] und der Managementsoftware, im *User Space*, zu ermöglichen, wird gemäß dem Konzept aus *Abschnitt 3.3.4* eine Schnittstellenkomponente geschaffen, die diese Funktionalität kapselt. Diese Schnittstellenkomponente kapselt die Treiberspezifischen Funktionen zu einer einheitlichen Schnittstelle *(Interface)* für die Managementsoftware.

Wie in *Abschnitt 3.2* beschrieben muss diese Schnittstellenkomponente eine Ereignisverarbeitung, als auch eine Interaktion mit dem Treiber ermöglichen. Beide Aufgaben werden bereits, für den verwendeten Treiber MadWifi, in der Komponente *driver_madwifi* des *hostapd* realisiert. Für die Interaktion mit MadWifi werden *Input/Output Controls* auf Basis der Systemfunktion *ioctl* [CRK05], sowie *Sockets* verwendet. So werden *ioctl*-Aufrufe dazu genutzt, Funktionen von MadWifi aus der Managementsoftware heraus aufzurufen, wie beispielsweise das Setzen eines Schlüssels oder eines Kanals, während über *Sockets* der asynchrone Austausch der EAPOL-Key-Frames zur Authentifizierung, sowohl von MadWifi zur Managementsoftware als auch von der Managementsoftware zu MadWifi realisiert wird.

Bei der Findung eines Mechanismus, der die von MadWifi eingehenden EAPOL-Key-Frames schließlich der richtigen Instanz zuordnet stellte es sich als hinderlich heraus, dass das Interface des *hostapd* inkompatibel zum Interface des *wpa_supplicant* ist. Um für dieses Implementierungsproblem eine prototypengerechte Lösung zu erzielen, weicht die Implementierung in diesem Punkt vom in *Abschnitt 3.3.4* dargelegten Konzept geringfügig ab. Abbildung 4.2 zeigt schematisch die umgesetzte Implementierung.

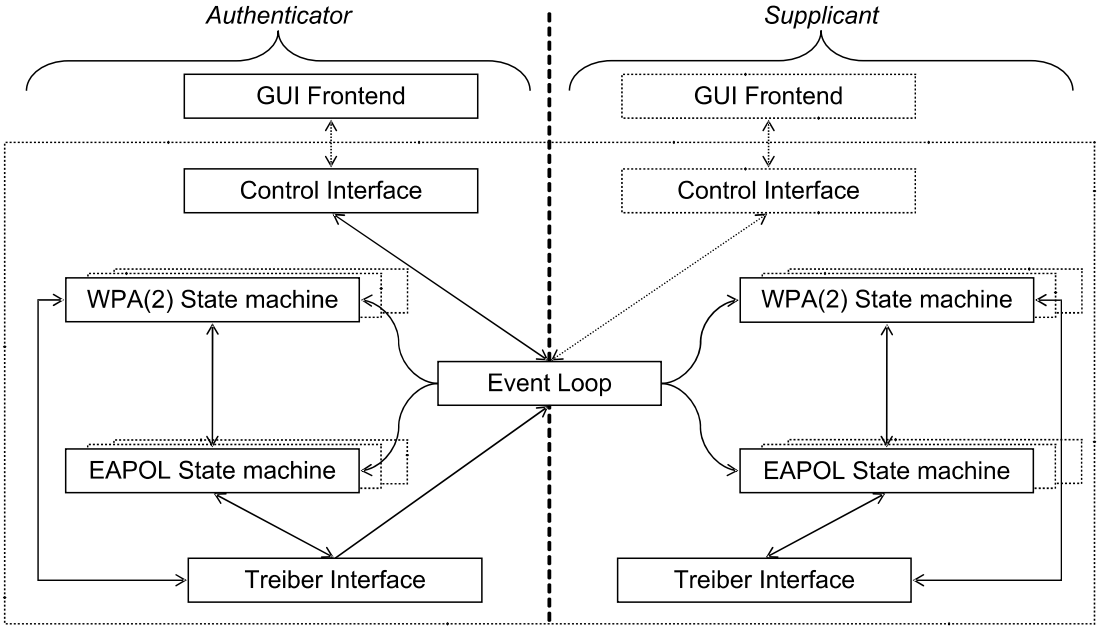

Abbildung 4.2: Struktogramm – Managementsoftware (Implementierung)

[1] Mehrheitlich im Kernel Space realisiert

Anders als im Konzept vorgesehen, werden die beiden *Treiber Interfaces* nicht zu einem einzigen *Treiber Interface* verschmolzen, sondern bleiben weiterhin getrennt. Damit bleiben die Interfaces zu den Authenticator-Instanzen bzw. zu den Supplicant-Instanzen ebenfalls unangetastet und Anpassungen, die ein einheitliches Interface nach sich zieht, entfallen. Funktionell ergeben sich aus dieser Maßnahme keine Nachteile gegenüber dem Konzept aus *Abschnitt 3*. Vielmehr lassen sich etwaige neue *Treiber Interfaces* des *hostapd* und des *wpa_supplicant* leichter portieren. In erster Linie muss bei einer Portierung die Ereignisverarbeitung, innerhalb des *Treiber Interfaces* des Supplicants, unterbunden, sowie auf Seiten des Authenticators für den Supplicant hinzugefügt werden. Im *Treiber Interface* des Authenticators kann so ein Dispatcher, auf Basis der Absender MAC-Adresse die Zuordnung der EAPOL-Key-Frames zur entsprechenden Authenticator-Instanz vornehmen, während im Supplicant jede Supplicant-Instanz sich innerhalb des *Treiber Interfaces* separat registriert. Eine Zuordnung über einen Dispatcher, analog zum *Treiber Interfaces* des Authenticators, ist nicht möglich, da das *Treiber Interface* des Supplicants keinen Zugriff auf die Verwaltungsdatenstruktur des Authenticators besitzt, um die Asynchronität zu erhalten.

Zur Realisierung der Ereignisverarbeitung im *Treiber Interface* des Authenticators wird, analog zu den vorangegangenen EAPOL-Key-Frames, ein *Socket* verwendet. Dieser ist vom Typ *RTNetlink Socket* und bildet im Rahmen der Interprozess-Kommunikation *(IPC)* [Tan03] zwischen MadWifi und der Managementsoftware eine Schnittstelle zur asynchronen Transferierung der in MadWifi erzeugten Ereignisse, in die Managementsoftware. Mit der Verwendung von *RTNetlink Sockets* definiert sich ebenfalls der Aufbau eines Ereignis-Paketes. So muss in jedem Paket ein Header *(struct nlmsghdr)* enthalten sein, der beispielsweise den Ereignistyp und die Prozess ID des Absenders enthält. Für die Managementsoftware relevant ist die Ereignisklasse der *Wireless Events*. Diese umfasst Ereignistypen, die speziell bei der Verwendung von drahtlosen Geräten auftreten können. Zwei dieser *Wireless Events* sind für die Managementsoftware von Bedeutung. Das erste ist vom Typ *RTM_NEWLINK* und entspricht einem *New Station*-Ereignis, während das zweite vom Typ *RTM_DELLINK* ist und dem *Station Leave*-Ereignis entspricht. Trifft eines dieser beiden Ereignisse ein, so muss die Managementsoftware darauf entsprechend reagieren. In *Abschnitt 4.2.3* wird daher, die Realisierung der Verarbeitung dieser Ereignisse näher betrachtet.

4.2.3 Ereignisverarbeitung

Nachdem im vorangegangenen Abschnitt beschrieben wurde, wie die Kommunikation mit dem Treiber erfolgt, beschreibt dieser Abschnitt die Verarbeitung dieser Ereignisse, sowie die, der zeitbasierten Ereignisse.

Für das Auslösen von Ereignissen innerhalb der Managementsoftware ist, dem Konzept entsprechend, die Komponente *Event Loop (eloop)* verantwortlich. Diese wird in unveränderter Form aus dem *hostapd* übernommen. Eine Funktion zur Ereignisverarbeitung muss sich so in der *Event Loop* unter Zuordnung ihres Kontextes

registrieren, so dass die *Event Loop* diese Funktion im Bedarfsfall benachrichtigen kann. Die Realisierung dieses Mechanismus erfolgt über Funktionszeiger der Programmiersprache C. Dabei erhält eine Ereignisverarbeitende Funktion eine festgelegte Parameterliste beim Aufruf. Bei Funktionen, zur Verarbeitung von zeitbasierten Ereignissen enthält der erste Parameter einen Kontextzeiger und der zweite einen optionalen Datenzeiger. Diese Funktionen zur Verarbeitung von zeitbasierten Ereignissen werden über einen *eloop_register_timeout*-Aufruf in der *Event Loop* registriert und somit einer Prioritäts-Queue hinzugefügt, wobei das relativ zum aktuellen Zeitpunkt am nächsten liegende Ereignis die höchste Priorität erhält. Ist der entsprechende Zeitpunkt erreicht, so wird die registrierte Funktion zusammen mit den Daten aufgerufen und aus der Prioritäts-Queue entfernt.

Die übrigen, vom Treiber ausgelösten Ereignisse, werden mit Hilfe der *select*-Funktion über den *RTNetlink Socket* empfangen und einer, zuvor mit einem *eloop_register_sock*-Aufruf registrierten Verarbeitungsfunktion, übermittelt. Anders als die zeitbasierten Funktionen zur Verarbeitung von Ereignissen wird zusätzlich zum Kontextzeiger und zum optionalen Datenzeiger die Socketnummer der Quelle, als erster Parameter, übermittelt. Bei der Verwaltung sind alle Funktionen als gleichwertig anzusehen, so dass eine verkettete Liste zur Verwaltung der Funktionen in der *Event Loop* genutzt wird. Wird ein Ereignis aus dem *Treiber Interface* registriert, so werden die Funktionen entsprechend in der Reihenfolge ihrer Registrierung aufgerufen.

Innerhalb der Managementsoftware gibt es, dem Konzept entsprechend, nur eine Funktion *(madwifi_wireless_event_receive)* zur Verarbeitung der Ereignisse aus dem Treiber, im *Treiber Interface* des Authenticators. Innerhalb dieser Funktion wird auf Basis der Absender MAC-Adresse auf das Ereignis reagiert. Ist das Ereignis vom Typ *Station Leave* so wird die zur MAC-Adresse korrespondierende Authenticator- und Supplicant-Instanz gelöscht. Wenn das Ereignis hingegen vom Typ *New Station* ist, so wird eine Authenticator- und Supplicant-Instanz, dem nachfolgenden Abschnitt entsprechend, erzeugt.

4.2.4 Instanzverwaltung

In der Umsetzung der Managementsoftware für den Ad-hoc-Modus ist es notwendig, dass für jede Gegenstelle sowohl eine Authenticator-Instanz als auch eine Supplicant-Instanz bereitgestellt wird. Da der *hostapd* bereits im Infrastruktur-Modus einen Mechanismus zur Verwaltung von Gegenstellen implementiert, wird dieser so erweitert, dass er ebenfalls die Supplicant-Instanzen verwalten kann.

Zur Verwaltung der Instanzen und weiteren notwendigen Informationen, wie z.B. die unterstützte Übertragungsrate, wird eine verkettete Liste genutzt. Um dennoch einen effizienten Zugriff zu Gewährleisten, wird eine Hash-Tabelle mit Platz für 256 Einträge in Kombination mit dem niedrigsten Byte der MAC-Adresse in kanonischer Darstellung als Hash-Wert verwendet. Eine neue Instanz wird analog zum Infrastruktur-Modus

erzeugt, wenn ein *New Station*-Ereignis durch den Treiber ausgelöst wird. Allerdings kann es, wie in Abbildung 4.3 dargestellt, bedingt durch die gegenseitige paarweise Authentifizierung im Ad-hoc-Modus dazu kommen, dass eine der beiden Stationen bereits mit der Authentifizierung beginnt, während die zweite noch keine Instanzen erzeugt hat.

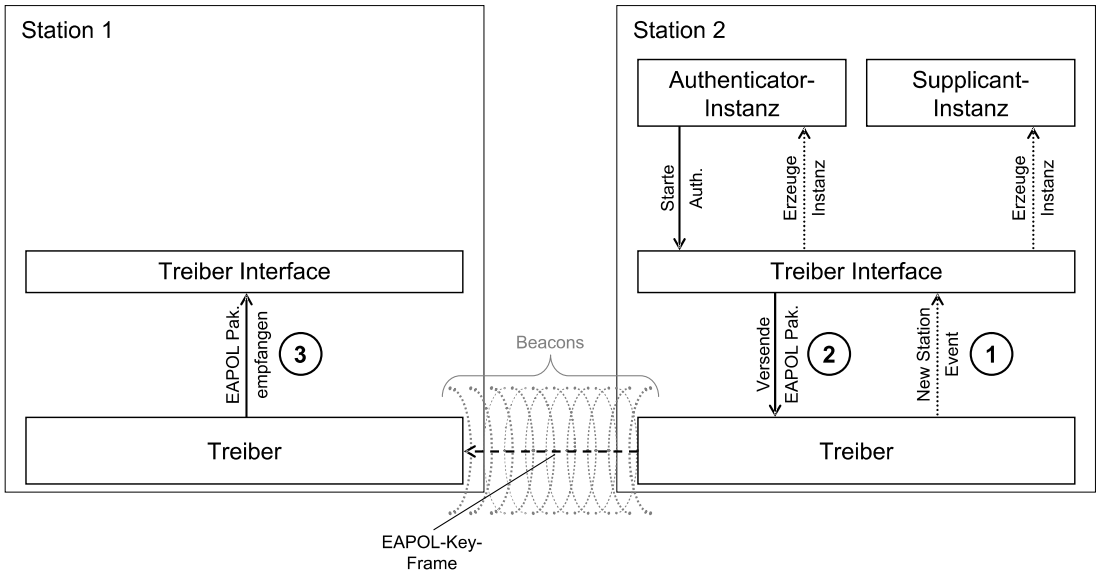

Abbildung 4.3: Ablauf einer asynchronen Authentifizierung

Zunächst wird die Managementsoftware, die ebenfalls bei Bedarf den Treiber startet und benötigte Konfigurationen über *ioctl* vornimmt, auf den beiden betrachteten Stationen gestartet, so dass beide Stationen *Beacon*-Frames versenden. Diese enthalten analog zum Infrastruktur-Modus ein WPA-Informationselement *(WPA_IE)* oder RSN-Informationselement *(RSN_IE)*, welches Auskunft über die unterstützten Security-Mechanismen der jeweiligen Station gibt. Sind die unterstützten Security-Mechanismen kompatibel, so erhält das *Treiber Interface* ein *New Station*-Event *(Zeitpunkt 1)*. Das *Treiber Interface* sorgt daraufhin dafür, dass eine neue Authenticator- und Supplicant-Instanz erzeugt wird, falls bis zu diesem Zeitpunkt keine existiert. Die Zustandsmaschinen werden anschließend initialisiert und das erste EAPOL-Key-Frame wird von der Authenticator-Instanz erzeugt. Das *Treiber Interface* sorgt dafür, dass dieses an den Treiber weiter geleitet wird *(Zeitpunkt 2)*. Nachdem der Treiber das EAPOL-Key-Frame versendet hat, muss dieses analog zum 4-Wege-Handshake im Infrastruktur-Modus von der Supplicant-Instanz der Gegenstelle beantwortet werden, andernfalls wird die Authentifizierung nach kurzer Zeit[1] eingestellt.

Kommt es allerdings aufgrund von Verzögerungen dazu, dass eine Station mit der Authentifizierung beginnt, während die Gegenstelle ihrerseits noch keine Supplicant-Instanz erzeugt hat, könnten die Frames nicht beantwortet werden und die Authentifizierung würde fehlschlagen. Deshalb wird in der Managementsoftware

[1] Größenordnung liegt in der Managementsoftware bei drei Sekunden

ebenfalls ein internes *New Station*-Ereignis ausgelöst, wenn ein kompatibles EAPOL-Key-Frame von einer nicht Authentifizierten Gegenstelle empfangen wird *(Zeitpunkt 3)*.

Um den Vorgang der Erzeugung der Authenticator- und Supplicant-Instanzen weiterhin speichereffizient zu gestalten, werden Informationen, die für jede Gegenstelle individuell sind, in der verwendeten Datenstruktur von Informationen, die für alle Gegenstellen benötigt werden, getrennt. Zu den letzteren zählen im Wesentlichen die Rückruffunktionen und die Konfigurationsinformationen. Diese Informationen werden bereits während der Initialisierung, jeweils für den Authenticator und für den Supplicant in einer Datenstruktur abgelegt, so dass bei der Erzeugung der jeweiligen Instanzen lediglich darauf zurückgegriffen werden muss. Da Funktionszeiger für die Rückruffunktionen den gleichen Speicherbedarf, wie eine neue Referenz benötigen, werden diese im Hinblick auf die Speichereffizienz lediglich kopiert, während auf die Konfigurationsinformationen Speichereffizient über eine Referenz zugegriffen werden kann. Tabelle 4.2 fasst die wichtigsten Elemente, der für alle Instanzen benötigten Informationen und deren Bedeutung zusammen.

Element	**Bedeutung**
struct wpa_global	Konfigurationsparameter (PSK, Verschlüsselungsprotokoll, …) der Supplicant-Instanzen
struct wpa_driver_ops_supp	*Treiber Interface* der Suplicant-Instanzen
ifname	Name des Interfaces
own_addr	Eigene MAC-Adresse
struct hostapd_config	Konfigurationsparameter (PSK, Verschlüsselungsprotokoll, …) der Authenticator-Instanzen
struct wpa_driver_ops	*Treiber Interface* der Authenticator-Instanzen

Tabelle 4.2: Statische Elemente

Bei den individuell benötigten Informationen handelt es sich im Wesentlichen um alle Zustandsinformationen, die für die Implementierung der Zustandsmaschinen benötigt werden. Diese werden auf zwei unterschiedliche Arten erzeugt. Für den Authenticator wird der bereits implementierte Mechanismus des Infrastruktur-Modus verwendet, bei dem jegliche dynamische Information separat erzeugt wird. Der Supplicant realisiert dies hingegen mit Hilfe von Super-Informationen, analog der Objektorientierten Programmierung. Es werden alle für die Zustandsmaschinen benötigten Informationen analog zum Infrastruktur-Modus, während der Initialisierung erzeugt. Nach der Erzeugung dieser Informationen ist eine voll funktionsfähige Supplicant-Instanz entstanden. Jedoch wird diese nicht verwendet, sondern bildet eine Eltern-Instanz von der die verwendeten Supplicant-Instanzen abgeleitet werden. Die Ableitung entspricht in der Implementierung einer Kopie der Eltern-Instanz, mit einer anschließenden Aktualisierung der verwendeten Referenzen. So wird ein großer Teil des benötigten

Aufwandes zur Erzeugung von Supplicant-Instanzen in die Initialisierungsphase der Managementsoftware verlagert.

4.3 Managementsoftware mit WMN Unterstützung

Damit die Managementsoftware zusätzlich zum IEEE 802.11i Standard auch für den WMN-Modus konfiguriert werden kann, ist es notwendig die Parameterliste aus *Abschnitt 4.2.1* um zwei Parameter zu erweitern. So wird ein Parameter *-M* hinzugefügt, der die Managementsoftware im Master-Modus startet und ein Parameter *-C* für den Client-Modus. Beide Parameter, sowie der Parameter *-A* für den Ad-hoc-Modus, schließen einander gegenseitig aus. Weitere bedeutende implementierungsspezifische Anpassungen werden nachfolgend betrachtet.

4.3.1 Aufbau der Broadcast-Nachrichten

Um den bereits authentifizierten Stationen mitzuteilen, dass sich eine neue Station mit dem WMN authentifizieren will, muss die nicht authentifizierte Station in regelmäßigen Abständen Broadcast-Nachrichten versenden. Einen analogen Mechanismus sieht das Konzept für die Re-Authentifizierung vor, wobei diese Nachricht sich von der, der Authentifizierung unterscheiden muss.

In der Implementierung könnten für diese Aufgabe zwei verschiedene Daten-Frames festgelegt werden, die im Datenteil lediglich ein Byte enthalten. Das Byte beinhaltet dann die Information, ob es sich bei dem Daten-Frame um eine Anfrage zur Authentifizierung oder um eine Anfrage zur Re-Authentifizierung, seitens der nicht authentifizierten Station, handelt. Allerdings muss der Treiber dieses *Daten*-Frame an die Managementsoftware übermitteln, so dass die Managementsoftware gegebenenfalls die Authentifizierung bzw. die Re-Authentifizierung initiieren kann. Da dieses definierte Daten-Frame weder MadWifi noch anderen Treibern bekannt sein würde, müsste dies erst in den Treibern implementiert werden. Abhilfe schafft die im IEEE 802.1X Standard eingeführte Kapselung von Daten für die Authentifizierung, in *EAPOL*-Frames. Diese *EAPOL*-Frames werden ebenfalls im IEEE 802.11i Standard in den Handshakes verwendet, so dass MadWifi und andere Treiber bereits über Mechanismen der Weiterleitung verfügen. Ausgehend von der im IEEE 802.11X Standard definierten Kapselung der Daten, zeigt Abbildung 4.4 die Einbettung der für die Authentifizierung und Re-Authentifizierung benötigten Informationen im WMN-Modus.

Der *PAE (Port Access Entity) Ethernet Type* mit dem definierten Wert 888Eh gibt an, dass es sich bei dem Frame, um ein Frame für die Port-basierte Authentifizierung, nach dem IEEE 802.1X Standard handelt. Im darauf folgenden Feld befindet sich die verwendete Protokollversion *(Protocol Version).* Die verwendete Protokollversion der Managementsoftware kann durch den Benutzer festgelegt werden, wobei der IEEE

802.1X Standard die Verwendung der Version zwei empfiehlt. Der Pakettyp *(Packet Type)* legt schließlich fest, welche Nutzdaten das Paket enthält. Beispielsweise entspricht der Wert *0000 0011* einem EAPOL-Key-Frame. Für die Anfrage zur Authentifizierung bzw. Re-Authentifizierung wird der Wert *0000 0000* verwendet, was einem EAP-Frame entspricht und dessen Aufbau in der Abbildung 4.4 weiter verfolgt wird. Weitere Pakettypen können [IEE04X] entnommen werden.

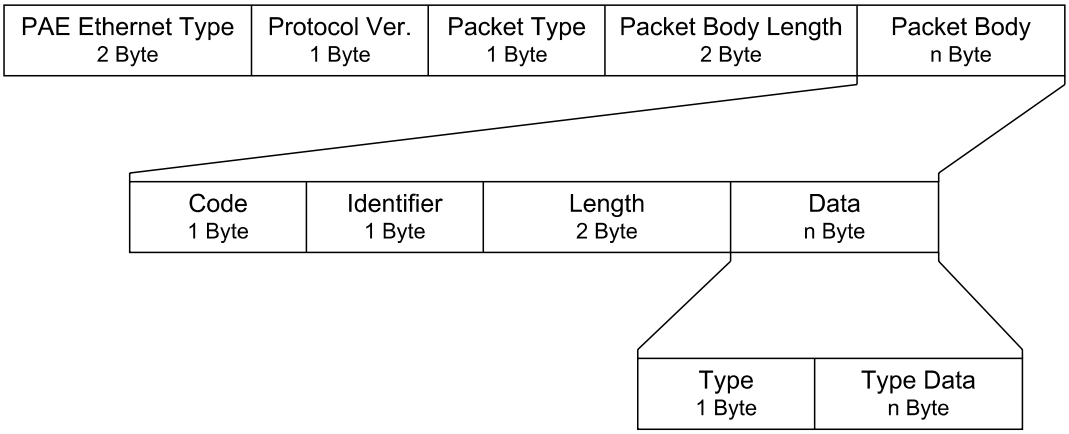

Abbildung 4.4: Einbettung der Re- / Authentifizierungsanfragen

Die Paketlänge *(Packet Body Length)* beinhaltet die Länge der eingebetteten Daten in Network Byte Order *(Big Endian)*. Abschließend folgen die gekapselten Daten *(Packet Body)*, die für Authentifizierung bzw. Re-Authentifizierungsanfragen ein EAP-Frame enthalten. Das EAP-Frame seinerseits beinhaltet zunächst einen *Code* der angibt, was die Aufgabe des EAP-Frames ist. Hier wird der Wert eins verwendet. Dieser gibt an, dass es sich um eine Anfrage *(Request)* handelt. Weitere mögliche Werte sind zwei für eine Antwort *(Response)*, drei für eine erfolgreiche Bestätigung *(Success)*, sowie vier für einen Fehler *(Failure)*. Sowohl für eine Anfrage, als auch für eine Antwort enthält das EAP-Frame einen Typen *(Type)* und die zugehörigen Typ-Daten *(Type Data)* im Datenfeld *(Data)*. Auf das *Code*-Feld folgt eine Identifizierung *(Identifier)* der zur Unterscheidung von verschiedenen Anfragen genutzt wird. Die zu einer Anfrage gehörende Antwort beinhaltet die gleiche Identifizierung. Da für die Authentifizierungs- bzw. Re-Authentifizierungsanfrage keine explizite Antwort gesendet wird, muss an dieser Stelle keine Unterscheidung der Identifizierung getroffen werden. Anschließend folgt eine Längenangabe *(Length)* des gesamten EAP-Frames, bevor in den Daten *(Data)* ein neuer Typ *(Type)* für die Anfrage definiert wird. Für die Authentifizierungs- bzw. Re-Authentifizierungsanfrage wird der Typ *82h* festgelegt. Weitere standardisierte Typen können [NWG04] entnommen werden. Nach dem Typ folgen abschließend die spezifischen Typ-Daten *(Type Data)*. Diese haben für den Typ *82h* eine feste Länge von einem Byte. Dieses eine Byte wird als Bitmaske verwendet, so dass bis zu acht Schlüssel angefragt werden können. Ein nicht gesetztes Bit bedeutet dabei, dass der Schlüssel mit diesem Index bereits bekannt ist. Bei einer Authentifizierungsanfrage sind entsprechend alle Bits gesetzt.

Durch diese Kapselung der Authentifizierungs- bzw. Re-Authentifizierungsanfrage in einem EAP-Frame, welches selbst Teil eines EAPOL-Frames ist, müssen an dieser Stelle keine speziellen Anpassungen in MadWifi oder anderen Treibern für den WMN-Modus vorgenommen werden. Die EAPOL-Frames werden unverändert durch die Treiber an die Managementsoftware weitergereicht, die sich um die Extrahierung und Verarbeitung der Informationen kümmert.

4.3.2 Erweiterung der Handshakes

Im WMN-Modus muss der Authenticator, sowohl bei der Authentifizierung als auch bei der Re-Authentifizierung, neben dem GK, auch die relative zeitliche Gültigkeit des GKs dem Supplicant mitteilen. Die Erzeugung des GKs und des GTKs sind äquivalent in der Managementsoftware realisiert, so dass der GK der Stärke des GTKs in nichts nachsteht. Zur Verteilung des GKs lässt sich ebenfalls der im IEEE 802.11i Standard spezifizierte Group Key-Handshake oder optional die dritte Nachricht des 4-Wege-Handshake nutzen. Damit muss für die Verteilung des GKs kein eigenständiges Protokoll implementiert werden.

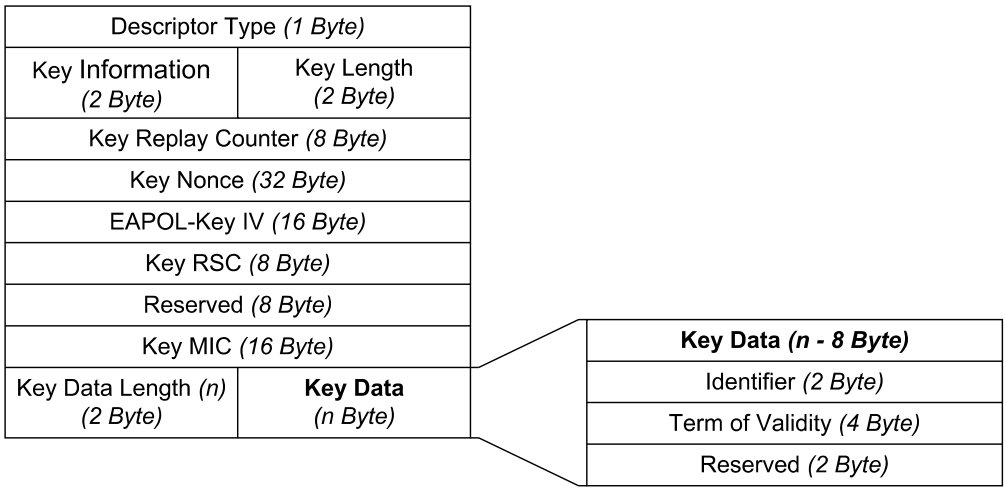

Abbildung 4.5: Erweiterung des EAPOL-Key Frames

Allerdings sieht der IEEE 802.11i Standard in den für die Verteilung von Schlüsseln genutzten EAPOL-Key-Frames keine Information der relativen zeitlichen Gültigkeit des Schlüssels vor. Diese Information könnte der Authenticator dem Supplicant in einem zusätzlichen Frame mitteilen. Dies hätte den Nachteil, dass zum einen eine Entkopplung zwischen dem Schlüssel und der relativen zeitlichen Gültigkeit des Schlüssels nötig wäre und zum anderen ein weiteres Frame gesendet werden müsste. Beide Nachteile würden weitere Probleme, wie z.B. die Zuordnung zwischen dem Schlüssel und der relativen zeitlichen Gültigkeit, die Gewährleistung der Security oder die Implementierung von Mechanismen zur Fehlerbehandlung, im Falle eines Verlustes der Konnektivität, während der zusätzlichen Kommunikation nach sich ziehen. Daher ist es besser den Schlüssel und dessen relative zeitliche Gültigkeit zusammen in einem Frame

zu verteilen. Da der IEEE 802.11i Standard in den EAPOL-Key-Frames eine Einbindung von Proprietären Schlüsselspezifischen Informationen erlaubt, bietet sich diese Möglichkeit als Alternative an. In Abbildung 4.5 ist die Einbettung der relativen zeitlichen Gültigkeit in ein EAPOL-Key-Frame dargestellt.

Der *Descriptor Type* gibt den genutzten Security-Mechanismus an. Dies entspricht in der Implementierung *1* für *RC4*, *2* für *RSN / WPA2* oder *254* für *WPA*. Mit Ausnahme von WPA sind diese in [IEE04X] definiert. Für den WMN-Modus kann sowohl *RSN / WPA2*, als auch *WPA* verwendet werden. Anschließend folgen Schlüsselinformationen *(Key Information)* und die Schlüssellänge *(Key Length)*. In den Schlüsselinformationen sind Informationen, wie z.B. der verwendete Verschlüsselungsalgorithmus oder ob es sich um ein Frame des 4-Wege-Handshakes oder eines des Group Key-Handshakes handelt enthalten. Als Schlüssellänge ist im WMN-Modus lediglich *16* erlaubt. Dies entspricht bei der Verwendung des CCMP-Verschlüsselungsprotokolls die Schlüssellänge. Der IEEE 802.11i Standard erlaubt an dieser Stelle zusätzlich *32* für TKIP, *5* für WEP-40 und *13* für WEP-104. Der *Key Replay Counter* ist ein Zähler zum Schutz vor Replay-Angriffen. Dieser wird mit Null initialisiert und bei jedem EAPOL-Key-Frame durch den Authenticator erhöht. Der Supplicant verwendet entsprechend den gleichen *Key Replay Counter*-Wert für eine Antwort auf ein empfangenes EAPOL-Key-Frame. Der *Key Nonce* wird lediglich zur Übertragung des ANonce und SNonce während der Authentifizierung, sowie Re-Authentifizierung verwendet. Im *EAPOL-Key IV* wird der IV abgelegt, der zur Ver- und Entschlüsselung des Frames verwendet wird. Das *Key RSC* dient unter anderem zur Synchronisierung des IEEE 802.11 Replay-Zustands. Auf den *Key RSC* folgt der *Key MIC*. Dieser wird zur Sicherung der Integrität des gesamten EAPOL-Key-Frames verwendet. Am Ende des EAPOL-Key-Frames folgen dann schließlich eine Information zur Länge *(Key Data Length)* der Schlüsseldaten und die Schlüsseldaten *(Key Data)* selbst. Zu den Schlüsseldaten wird im WMN-Modus, neben dem verschlüsselten GK, auch die relative zeitliche Gültigkeit des GKs hinzugefügt und verschlüsselt. Dazu werden die letzten acht Byte der Schlüsseldaten verwendet. Um sicher zu stellen, dass es sich tatsächlich um die relative zeitliche Gültigkeit des GKs handelt werden die ersten zwei Bytes *(Identifier)* zur Identifikation verwendet. Die verwendete Identifikation ist *2609h*. Darauf folgt die relative zeitliche Gültigkeit *(Term of Validity)* des GKs in Sekunden, ebenfalls in der Network Byte Order. Am Ende folgen bleiben zwei Bytes *(Reserved)* ungenutzt, die bisher noch nicht verwendet werden. Diese haben jeweils den Wert Null und können für zukünftige Erweiterungen genutzt werden.

4.3.3 Rückruffunktionen

Nachdem sich der Supplicant erfolgreich mit einem Authenticator aus dem WMN authentifiziert hat, muss dafür gesorgt werden, dass der erhaltene GK weiter verteilt werden kann. Da die für die Verteilung des GKs notwendigen Mechanismen im Authenticator implementiert sind, muss eine Schnittstelle zwischen dem Supplicant und

dem Authenticator geschaffen werden, die einen Informationsaustausch zwischen einer Supplicant-Instanz und einer Authenticator-Instanz gewährleistet.

Um weiterhin die Robustheit der Authentifizierung zu erhöhen, kann es in der Praxis sinnvoll sein mehrere Authentifizierungen parallel durchzuführen. So könnte beispielsweise aufgrund von Mobilität, die Konnektivität zwischen den Authentifizierungspartnern während der Authentifizierung verloren gehen, so dass die nicht authentifizierte Station erneut Authentifizierungsanfragen senden muss. Eine Möglichkeit dies, in einigen Situationen, zu vermeiden besteht darin, nicht nur auf die erste Antwort eines Authenticators zu reagieren, sondern auf mehr als einen Authenticator, beispielsweise auf drei oder auf beliebig viele, solang keine Authentifizierung erfolgreich verlief. Bricht in diesen Fällen eine Verbindung ab, so kann weiterhin die Authentifizierung, ohne erneutes versenden von Authentifizierungsanfragen, durchgeführt werden. Die Authentifizierung ist entsprechend dann erfolgreich durchgeführt, wenn mindestens eine der Authentifizierungen erfolgreich verlief.

Da die Managementsoftware bereits im IEEE 802.11i Standard mehrere Supplicant-Instanzen verwalten können musste, lässt sich dieser Mechanismus einfach in den WMN-Modus portieren. Allerdings müssten nicht alle Authentifizierung vollständig durchgeführt werden, so dass zwischen den Supplicant-Instanzen eine Kommunikation über eine zusätzliche Schnittstelle ermöglicht wird. Abbildung 4.6 stellt diese Erweiterung, gegenüber der Managementsoftware aus Abbildung 4.2 schematisch dar.

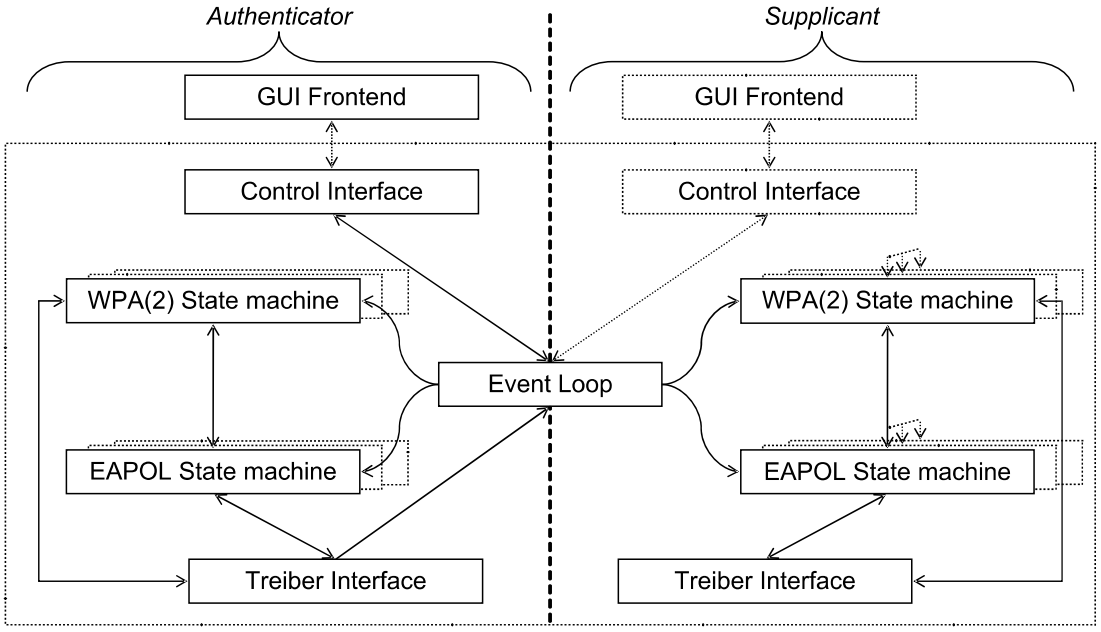

Abbildung 4.6: Struktogramm – Erweiterte Managementsoftware (Implementierung)

Die Umsetzung der beiden zuvor vorgestellten Schnittstellen erfolgt unter Nutzung von Rückruffunktionen. Rückruffunktionen werden in der Programmiersprache C als statische Funktionen implementiert, so dass sie über den Aufruf eines Funktionszeigers, der die Referenz auf die Funktion hält, aus einer beliebigen Codepassage aufgerufen

werden können. So ist es aus dem Kontext einer Supplicant-Instanz heraus möglich, dem Authenticator über den Aufruf eines Funktionszeigers, der eine Funktion im Authenticator-Kontext referenziert, den erhaltenen GK mitzuteilen und damit den Authenticator zu initialisieren, so dass andere Supplicants sich authentifizieren können.

In ähnlicher Weise wird dies auch zwischen den Supplicant-Instanzen realisiert. So teilt eine Supplicant-Instanz der Eltern-Supplicant-Instanz, über eine Rückruffunktion mit, dass die Authentifizierung erfolgreich verlief. Die Elter-Supplicant-Instanz sorgt anschließend dafür, dass keine neuen eingehenden EAPOL-Key-Frames beantwortet werden. Sollen darüber hinaus die zu diesem Zeitpunkt noch laufenden Authentifizierung abgebrochen werden, so müsste eine weitere Rückruffunktion im Authenticator-Kontext dazu genutzt werden, da dieser die entsprechenden Instanzen verwaltet und die nicht länger benötigten Supplicant-Instanzen aus der verwendeten Datenstruktur entfernt. Von dieser Möglichkeit macht die prototypische Implementierung allerdings keinen Gebrauch, da die Belastung des Mediums beim abbrechen einer nahezu vollendeten Authentifizierung, aufgrund der Versuche des erneuten Versendens, größer wäre, als die Beendigung der noch laufenden Authentifizierungen.

5 Evaluierung

In diesem Kapitel wird die Realisierung des IEEE 802.11i Standards für den Ad-hoc-Modus, sowie die speziell für WMNs entwickelte Lösung auf ihre Funktionsfähigkeit und Performance getestet. Durch Nutzung von speziellen Hardware-Komponenten der WLAN-Geräte lassen sich die Verschlüsselungsmechanismen effizient umsetzen, so dass die Verwendung der Security-Verfahren WEP und WPA-None einen vernachlässigbaren Einfluss auf die Leistungsfähigkeit eines WMNs hat. Daher kann an dieser Stelle auf eine weitere Untersuchung verzichtet werden.

Im Gegensatz zu WEP und WPA-None verfügen die entwickelten Verfahren, zur Erhöhung der Security, über ein automatisiertes Schlüsselmanagement und einen Authentifizierungsmechanismus. Dies verursacht gegenüber WEP und WPA-None einen erhöhten Kommunikationsaufwand, der beispielsweise die Zeit zwischen dem Einschalten einer Station und der Kommunikationsfähigkeit selbiger verlängert. Über eine gesamte Netzwerktopologie betrachtet, verlängert sich so auch die Zeit vom Einschalten aller Stationen bis zur vollen Konnektivität dieser Stationen *(Initiale Anlaufzeit)*, so dass sich durch eine Untersuchung der Initialen Anlaufzeit, Aussagen über die Kosten der Security-Mechanismen treffen lassen.

Zunächst wird für die entwickelten Verfahren die benötigte Initiale Anlaufzeit eines Netzwerks, an drei verschiedenen Testszenarien, untersucht. Einführend wird hierzu der Versuchsaufbau dargelegt und die erwarteten Ergebnisse diskutiert. Anschließend werden die Messergebnisse für den IEEE 802.11i Standard im Ad-hoc-Modus, gefolgt von denen im WMN-Modus vorgestellt, ausgewertet und schließlich miteinander verglichen.

Da der WMN-Modus auch während einer Re-Authentifizierung, eine unterbrechungsfreie Kommunikation *(siehe Abschnitt 3.4.4)* erlauben soll, wird dies in einem gesonderten Testszenario überprüft. Dazu wird das verwendete Testszenario beschrieben, bevor das Kapitel mit der Vorstellung und Auswertung der Messergebnisse schließlich abgeschlossen wird.

5.1 Initiale Anlaufzeit eines Netzwerks

5.1.1 Versuchsaufbau

Für den Aufbau von verschiedenen Szenarien standen acht stationäre Rechner, mit voller Konnektivität zueinander, zur Verfügung. Jeder Rechner besaß zur drahtlosen Kommunikation eine WLAN-Karte mit Atheros Chipsatz, so dass ein MadWifi-Treiber[1], mit den in *Abschnitt 4.1* beschriebenen Erweiterungen, verwendet werden konnte. Auf dieser Grundlage wurde ein Ad-hoc-Netzwerk auf der Frequenz 2437 MHz *(Kanal 6)* und einer Brutto-Datenrate von 54 MBit/s errichtet, sowie ein Pre-Shared Key

[1] Revision 3851

für die Security-Verfahren festgelegt. Um das Verhalten der Verfahren auch in Netzwerken mit vergleichsweise geringer Konnektivität zu untersuchen, wurden in der entwickelten Managementsoftware MAC-Filter eingebaut. Diese sorgen dafür, dass Frames von zuvor festegelegten Absender-Stationen ignoriert werden, wodurch eine Authentifizierung mit entsprechenden Stationen unterbunden werden kann.

Weiterhin wurden zur Bestimmung der Initialen Anlaufzeit des Netzwerks, zu Ereignissen in der Managementsoftware, Zeitstempel hinzugefügt. Für die Messungen relevant sind dabei das Starten der Managementsoftware, welche hierbei gleichermaßen die Station aktiviert, sowie der Zeitpunkt des Abschlusses des Schlüsselaustauschs[1]. Aus der Differenz dieser Zeitstempel ergibt sich die Zeit, die diese einzelne Station benötigt um kommunikationsfähig zu sein. Durch Verwendung des Network Time Protocols *(NTP)* soll gewährleistet werden, dass die lokalen Uhren aller Stationen im Netzwerk synchronisiert sind. Werden unter der Voraussetzung synchronisierter Uhren, alle Stationen im Netzwerk gleichzeitig gestartet, so ergibt sich die Initiale Anlaufzeit des gesamten Netzwerks aus dem Startzeitpunkt einer beliebigen Station und dem Zeitpunkt, über alle Stationen betrachtet, der letzten Fertigstellung eines Schlüsselaustauschs. Allerdings ist es praktisch unmöglich alle Stationen simultan zu starten und die Uhren wirklich exakt zu synchronisieren. Für die Experimente wurde daher ein Abgleich der Uhren vorgenommen und ein Script zum zeitnahen Starten der Stationen, in numerischer Reihenfolge, verwendet. So konnte für die Experimente, in der Summe, eine maximale Latenz von 100 ms erreicht werden.

Verfahren	Parameter	Wert
IEEE 802.11i	Zeitbeschränkung[2] einer Handshake-Nachricht[3]	1 s
	Maximale Anzahl von Zeitüberschreitungen	3
	Zeit zwischen erneuter Authentifizierung	1 s
	Beacon-Intervall	100 ms
WMN-Modus	Zeitbeschränkung einer Handshake-Nachricht	1 s
	Maximale Anzahl von Zeitüberschreitungen	3
	Zeit zwischen erneuter Authentifizierung	1 s
	Beacon-Intervall	100 ms
	Zeitintervall der Authentifizierungs-Anfragen	1 s
	Anzahl der Supplicant-Instanzen pro Station	3

Tabelle 5.1: Parameter zur Bestimmung der Initialen Anlaufzeit

Um weiterhin den Einfluss von externen Störungen auf die Messergebnisse zu reduzieren und so ein repräsentativeres Ergebnis zu erhalten, wurde jede Messung zwanzigmal wiederholt. Die übrigen, für die einzelnen Messungen, relevanten

1 Übermittlung des empfangenen GTKs an den Treiber
2 IEEE 802.11i Standard empfiehlt 100 ms; wurde bereits im *hostapd* auf 1 s verlängert
3 4-Wege-Handshake, wie auch Group Key-Handshake

Konfigurationsparameter, wurden während sämtlicher Messungen nicht verändert und sind in der Tabelle 5.1 zusammenfassend aufgeschlüsselt.

Sämtliche Messungen zur Bestimmung der Initialen Anlaufzeit des Netzwerkes wurden an drei Testszenarien, sowie mit einer variierenden Anzahl von Stationen, ermittelt. Die Testszenarien basieren auf drei verschiedenen Netzwerktopologien, die nachfolgend vorgestellt werden. Zusätzlich werden die erwarteten Ergebnisse für die entwickelten Verfahren diskutiert.

Ketten-Topologie

In der Ketten-Topologie verfügt jede Station, mit Ausnahme der Station 1 und 8, über zwei Nachbarstationen. Abbildung 5.1 veranschaulicht diese Topologie.

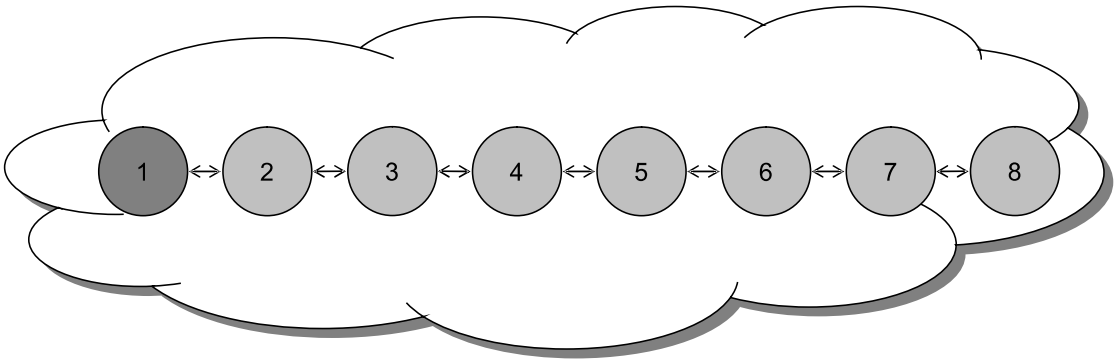

Abbildung 5.1: Ketten-Topologie

Bei der Verwendung des IEEE 802.11i Standards für den Ad-hoc-Modus müssen sich Station 1 und Station 2, Station 2 und Station 3, Station 3 und Station 4, Station 4 und Station 5, Station 5 und Station 6, Station 6 und Station 7, sowie Station 7 und Station 8 jeweils paarweise, mit einem 4-Wege-Handshake, authentifizieren. Darüber hinaus muss jede Station für jede authentifizierte Station einen Group Key-Handshake initiieren, um den eigenen Gruppenschlüssel zu verteilen und so eine Multicast-Verbindung zu ermöglichen. Somit muss jede Station, mit jeder Gegenstelle, vier Nachrichten für einen 4-Wege-Handshake, gefolgt von zwei Nachrichten für einen Group Key-Handshake austauschen, um der Gegenstelle eine spätere Kommunikation zu ermöglichen. Der benötigte Nachrichtenaustausch beläuft sich somit auf sechs Nachrichten. Zur Vereinfachung der weiteren Betrachtungen, wird der 4-Wege-Handshake, gefolgt von einem Group Key-Handshake fortan als *Handshake* bezeichnet.

Da sämtliche Stationen über ein gemeinsames physikalisches Medium kommunizieren, kann zu einem Zeitpunkt lediglich eine Nachricht versendet werden, so dass während dieser Zeit andere Stationen keine Nachrichten senden können. Unter der Berücksichtigung der empfohlenen Zeitbeschränkung des IEEE 802.11i Standards, von 100 ms, für die Beantwortung einer gesendeten Nachricht, kann die Annahme getroffen

werden, dass ein vollständiger Handshake maximal 300[1] ms und so im Mittel 150 ms benötigt. Damit ist das Medium für etwa 150 ms, durch einen Handshake, blockiert.

Bevor jedoch der Handshake durchgeführt werden kann, muss eine Zelle etabliert werden. Da das Beacon-Intervall für die Messungen auf 100 ms *(siehe Tabelle 5.1)* festgelegt ist, wird unabhängig von der Anzahl der Stationen für die Etablierung der Zelle eine Zeit von 100 ms einkalkuliert. Aus der Zeit, zur Etablierung einer Zelle, sowie aus der Summe der Zeiten, der einzelnen Handshakes, lässt sich die Initiale Anlaufzeit für ein Netzwerk bestimmen. Für die Ketten-Topologie und den IEEE 802.11i Standard, sowie unter den zuvor dargelegten Zeiten lässt sich folgende Funktion in Abhängigkeit von der Anzahl der Stationen *(n)* zur Berechnung der Initialen Anlaufzeit definieren.

$$f(n_{\geq 2}) = 2 * (n - 1) * 150\ ms + 100\ ms$$

In der nachfolgendenden Tabelle 5.2 wird für die variierende Anzahl an Stationen in der Kettentopologie, die zuvor definierte Funktion genutzt, um eine Abschätzung der erwarteten Initialen Anlaufzeiten zu bestimmen. Zusätzlich wird die Anzahl der Handshakes, sowie die benötigte Anzahl der Nachrichten aufgeschlüsselt.

Anzahl der Stationen *(n)*	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anzahl der Handshakes	2	4	6	8	10	12	14
Anzahl der Nachrichten	12	24	36	48	60	72	84
Initiale Anlaufzeit [s]	0,4	0,7	1,0	1,3	1,6	1,9	2,2

Tabelle 5.2: Erwartete Anlaufzeiten des IEEE 802.11i in der Ketten-Topologie

Im WMN-Modus übernimmt die Station 1 die festgelegte Rolle des Masters. Station 2 muss sich so bei Station 1, Station 3 bei Station 2, Station 4 bei Station 3, Station 5 bei Station 4, Station 6 bei Station 5, Station 7 bei Station 6 und Station 8 bei Station 7 authentifizieren, um anschließend den GK zu erhalten. Damit müssen die Stationen 1 bis 7, den Handshake jeweils einmal initiieren.

Analog zum IEEE 802.11i Standard benötigt ein Handshake im Mittel 150 ms und die Etablierung einer Zelle etwa 100 ms. Allerdings spielen im WMN-Modus noch weitere Faktoren eine Rolle. So kann sich eine Station nur durch eine andere, bereits authentifizierte, Station authentifizieren lassen, so dass die Anzahl der abhängigen Authentifizierung mit in die Betrachtung aufgenommen werden muss. Somit handelt es sich bei der Ketten-Topologie um den Worst Case des WMN-Modus und gleichermaßen um den Best Case des IEEE 802.11i Standards.

Darüber hinaus muss jede Client-Station eine explizite Authentifizierungs-Anfrage senden, welche unter der Berücksichtigung der Parameter aus Tabelle 5.1, im Mittel

[1] Authenticator sendet Nachricht zum Zeitpunkt Null und erhält Antwort spätestens nach 100 ms. Damit sind zwei von sechs Nachrichten innerhalb von 100 ms abgeschlossen.

500 ms benötigt. Der letzte Unterschied betrifft die Erzeugung des Ereignisses zur Initiierung der Authentifizierung und damit der Handshakes. Beim IEEE 802.11i Standard werden die Ereignisse im Prinzip simultan, durch die Treiber der beteiligten Stationen, an die Managementsoftware gereicht, während im WMN-Modus der Treiber nur die Aufgabe hat, die von anderen Stationen erhaltende Authentifizierungs-Anfrage an die Managementsoftware weiter zu leiten. Bei Funktionstests hat sich gezeigt, dass die Socket-Verbindung zwischen dem Treiber und der Managementsoftware häufig die erste eingehende Authentifizierungs-Anfrage nicht korrekt an die Managementsoftware weiterleitet, so dass hieraus, unter der Berücksichtigung der Parameter aus Tabelle 5.1, eine Verzögerung der Initiierung von einer Sekunde resultiert. Diese fließt als Konstante in die nachfolgende Funktion zur Berechnung der Initialen Anlaufzeit der Kette-Topologie im WMN-Modus ein.

$$f(n_{\geq 2}) = (n - 1) * (150\ ms + 500\ ms) + 100\ ms + 1000\ ms$$

In der Tabelle 5.3 wird diese Funktion dazu genutzt, eine Abschätzung der Initialen Anlaufzeiten zu bestimmen. Zusätzlich werden die Anzahl der Handshakes, sowie die davon abhängigen Handshakes und die benötigten Nachrichten aufgeschlüsselt.

Anzahl der Stationen *(n)*	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anzahl der Handshakes	1	2	3	4	5	6	7
Abhängige Handshakes	1	2	3	4	5	6	7
Anzahl der Nachrichten	6	12	18	24	30	36	42
Initiale Anlaufzeit [s]	1,75	2,4	3,05	3,7	4,35	5,0	5,65

Tabelle 5.3: Erwartete Anlaufzeiten des WMN-Modus in der Ketten-Topologie

Topologie mit voller Konnektivität

Dies ist nach der vorangegangen Ketten-Topologie ein weiterer Extremfall. Bei dieser Topologie hat jede Station zu jeder anderen Station eine Verbindung. Abbildung 5.2 zeigt schematisch eine solche voll vermaschte Topologie.

Für den IEEE 802.11i Standard bedeutet eine solche Topologie, dass jede Station den Handshake siebenmal für seine Nachbarstationen initiieren muss. Dies entspricht dem, bereits in *Abschnitt 3.1* bestimmten, Worst Case des IEEE 802.11i Standards für Ad-hoc-Netzwerke. Die übrigen benötigten Parameter können, analog zur Ketten-Topologie verwendet werden, so dass sich für eine Topologie mit voller Konnektivität für den IEEE 802.11i Standard folgende Funktion zur Berechnung der Initialen Anlaufzeit herleiten lässt.

$$f(n_{\geq 2}) = (n^2 - n) * 150\ ms + 100\ ms$$

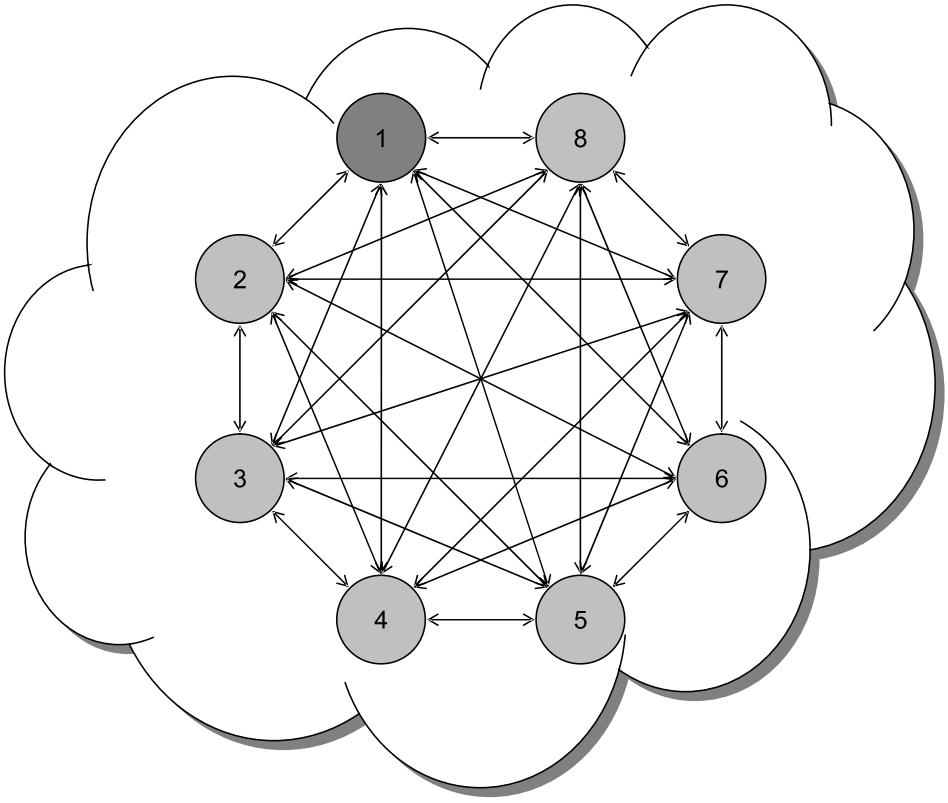

Abbildung 5.2: Topologie mit voller Konnektivität

Diese Funktion wird genutzt, um in Tabelle 5.4 das Erwartungsbild der Initialen Anlaufzeiten für die Messungen aufzuschlüsseln. Wie schon bei der Ketten-Topologie wird darüber hinaus die benötigte Anzahl an Handshakes, sowie die benötigte Anzahl von Nachrichten aufgeschlüsselt.

Anzahl der Stationen *(n)*	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anzahl der Handshakes	2	6	12	20	30	42	56
Anzahl der Nachrichten	12	36	72	120	180	252	336
Initiale Anlaufzeit [s]	0,4	1,0	1,9	3,1	4,6	6,4	8,5

Tabelle 5.4: Erwartete Anlaufzeiten des IEEE 802.11i bei voller Konnektivität

Für den WMN-Modus muss eine Station die Rolle des Masters einnehmen. Diese Aufgabe übernimmt in der Topologie aus Abbildung 5.2 die Station 1. Anders als bei der Ketten-Topologie, können die Handshakes in dieser Topologie, auch im WMN-Modus ohne Abhängigkeiten durchgeführt werden. Dies ist möglich, da sämtliche Stationen über Konnektivität zur Master-Station verfügen. Darüber hinaus stellt diese Topologie den Best Case für den WMN-Modus dar, steht aber zugleich im Widerspruch zu der Philosophie eines WMNs.

Wie schon in der Ketten-Topologie lässt sich für die Topologie mit voller Konnektivität eine Funktion zur Berechnung der Initialen Anlaufzeit angeben. Dazu werden die gleichen Werte, wie bei der Ketten-Topologie zugrunde gelegt.

$$f(n_{\geq 2}) = (n - 1) * 150\ ms + 500\ ms + 100\ ms + 1000\ ms$$

In der Tabelle 5.5 wird diese Funktion dazu genutzt, eine Abschätzung der Initialen Anlaufzeiten zu bestimmen.

Anzahl der Stationen *(n)*	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anzahl der Handshakes	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Abhängige Handshakes	0	0	0	0	0	0	0
Anzahl der Nachrichten	*6*	*12*	*18*	*24*	*30*	*36*	*42*
Initiale Anlaufzeit [s]	1,75	1,9	2,05	2,2	2,35	2,5	2,65

Tabelle 5.5: Erwartete Anlaufzeiten des WMN-Modus bei voller Konnektivität

Typische Topologie

Nachdem die Ketten-Topologie und die Topologie mit voller Konnektivität zwei Extremfälle abgebildet haben, sind die Stationen in dieser letzten Topologie nach typischer Art und Weise angeordnet. Das Resultat dieser Anordnung stellt Abbildung 5.3 schematisch dar.

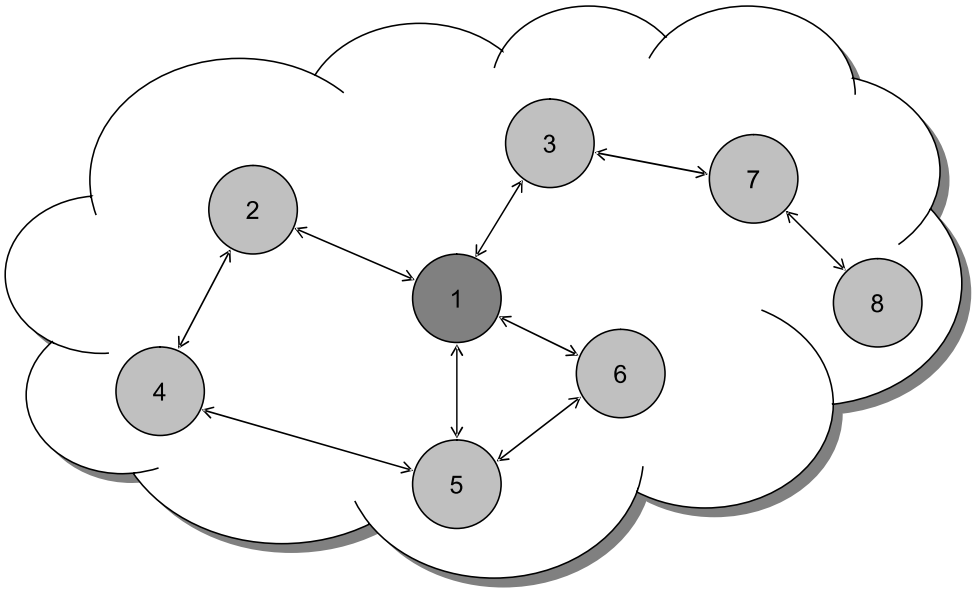

Abbildung 5.3: Typische Topologie

Der IEEE 802.11i Standard verlangt unabhängig von der Topologie immer eine paarweise Authentifizierung. Damit müssen die Handshakes, auch in dieser typischen Topologie, von beiden Kommunikationspartnern initiiert werden. Anders als bei der Ketten-Topologie und der Topologie mit voller Konnektivität lässt sich für diese typische Topologie nur schwer eine Funktion zur Berechnung der Initialen Anlaufzeit bestimmen. Daher wird die benötigte Anzahl an Handshakes, sowie die daraus resultierende Anzahl von Nachrichten lediglich in der Tabelle 5.6, zusammen mit den erwarten Initialen Anlaufzeiten aufgeschlüsselt.

Anzahl der Stationen *(n)*	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anzahl der Handshakes	2	4	6	10	14	16	18
Anzahl der Nachrichten	12	24	36	60	84	96	108
Initiale Anlaufzeit [s]	0,4	0,7	1,0	1,6	2,2	2,5	2,8

Tabelle 5.6: Erwartete Anlaufzeiten des IEEE 802.11i in einer typischen Topologie

Da der Kommunikationsaufwand sich gegenüber der Ketten-Topologie und Topologie voller Konnektivität im WMN-Modus nicht ändert, ist es für die Initiale Anlaufzeit von Bedeutung, ob Handshakes abhängig durchgeführt werden oder nicht. Mit Ausnahme der Stationen 4, 7 und 8 verfügen sämtliche Stationen über Konnektivität zur Master-Station. Station 4 kann sich mit Station 2 oder 5, erst nach deren erfolgreicher Authentifizierung authentifizieren. Station 7 kann sich analog erst nach erfolgreicher Authentifizierung der Station 3, bei selbiger authentifizieren und Station 8 gar erst im Anschluss. Wie schon beim IEEE 802.11i Standard lässt sich nur schwer eine Funktion zur Bestimmung der Initialen Anlaufzeit finden, so dass die Angaben ausschließlich in der Tabelle 5.7, zusammen mit den in etwa erwarteten Initialen Anlaufzeiten, angegeben werden.

Anzahl der Stationen *(n)*	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Anzahl der Handshakes	*1*	*2*	*3*	*4*	*5*	*6*	*7*
Abhängige Handshakes	0	0	1	1	1	2	3
Anzahl der Nachrichten	*6*	*12*	*18*	*24*	*30*	*36*	*42*
Initiale Anlaufzeit [s]	1,75	1,9	2,55	2,7	2,85	3,5	4,15

Tabelle 5.7: Erwartete Anlaufzeiten des WMN-Modus in einer typischen Topologie

5.1.2 Auswertung – IEEE 802.11i Standard

Nachfolgend werden die Messergebnisse, für die drei im vorangegangenen Abschnitt vorgestellten Topologien, unter Verwendung der Implementierung des IEEE 802.11i Standards für Ad-hoc-Netzwerke präsentiert. Dabei wird zu jeder Topologie die erwartete Initiale Anlaufzeit, die Messreihe mit der kleinsten Initialen Anlaufzeit, mit der größten Initialen Anlaufzeit, sowie die mittlere Initiale Anlaufzeit des Netzwerks gegenüber gestellt.

Ketten-Topologie

Wie der Abbildung 5.4 zu entnehmen ist, weisen die minimale und mittlere Initiale Anlaufzeit des Netzwerks ab einer Stationsanzahl von vier, einen geringen Anstieg auf. Für eine Stationsanzahl von fünf bis acht Stationen ist dies bei der maximalen Anlaufzeit ebenfalls ersichtlich. Dies bedeutet, dass mit einer zunehmenden Anzahl von Stationen auch die Initiale Anlaufzeit des gesamten Netzwerks nur leicht ansteigt. Eine Ausnahme stellen die Topologien mit einer Stationsanzahl von zwei, drei und vier bzw. bei der maximalen Anlaufzeit auch fünf dar. Hier ist ein stärkerer Anstieg im nahezu linearer Verlauf zu erkennen. Der Kurvenverlauf der minimalen Initialen Anlaufzeit verhält sich in etwas so, wie es im Vorfeld erwartet wurde. Die mittlere Initiale Anlaufzeit liegt jedoch deutlich darüber. Daraus lässt sich schließen, dass es während der überwiegenden Anzahl von Messungen zu Paketverlusten gekommen ist, die dazu führen, dass die Nachricht oder gar der komplette Handshake erneute durchgeführt werden musste.

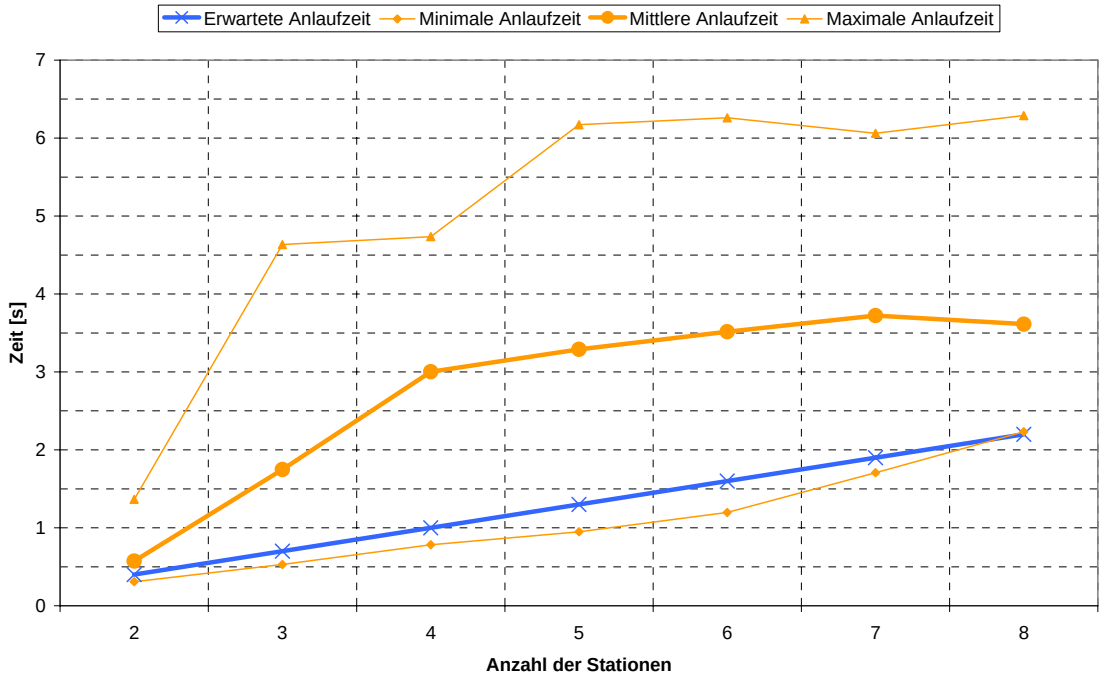

Abbildung 5.4: Initiale Anlaufzeit des IEEE 802.11i in der Ketten-Topologie

Topologie mit voller Konnektivität

Die in Abbildung 5.5 dargestellten Ergebnisse zeigen einen überwiegend linearen Verlauf der Initialen Anlaufzeiten, in Abhängigkeit zur Anzahl der Stationen. Dadurch, dass in dieser Topologie der Kommunikationsaufwand zur Authentifizierung mit einer zunehmenden Anzahl von Stationen quadratisch ansteigt, wurde ein leicht parabelförmiger Kurvenverlauf erwartet. Dennoch liegen die mittleren Initialen Anlaufzeiten von zwei, sieben und acht Stationen nur leicht oberhalb der erwarteten Größenordnung. Während der übrigen Messungen kam es zu höheren Paketverlusten, welche in der Vorbetrachtung unberücksichtigt blieben. Allerdings liegt die minimale Initiale Anlaufzeit, unabhängig von der Anzahl der Stationen, immer unterhalb der erwarteten Größenordnung. Dies bedeutet, dass während dieser Messung das Medium geringer als einkalkuliert belastet wurde.

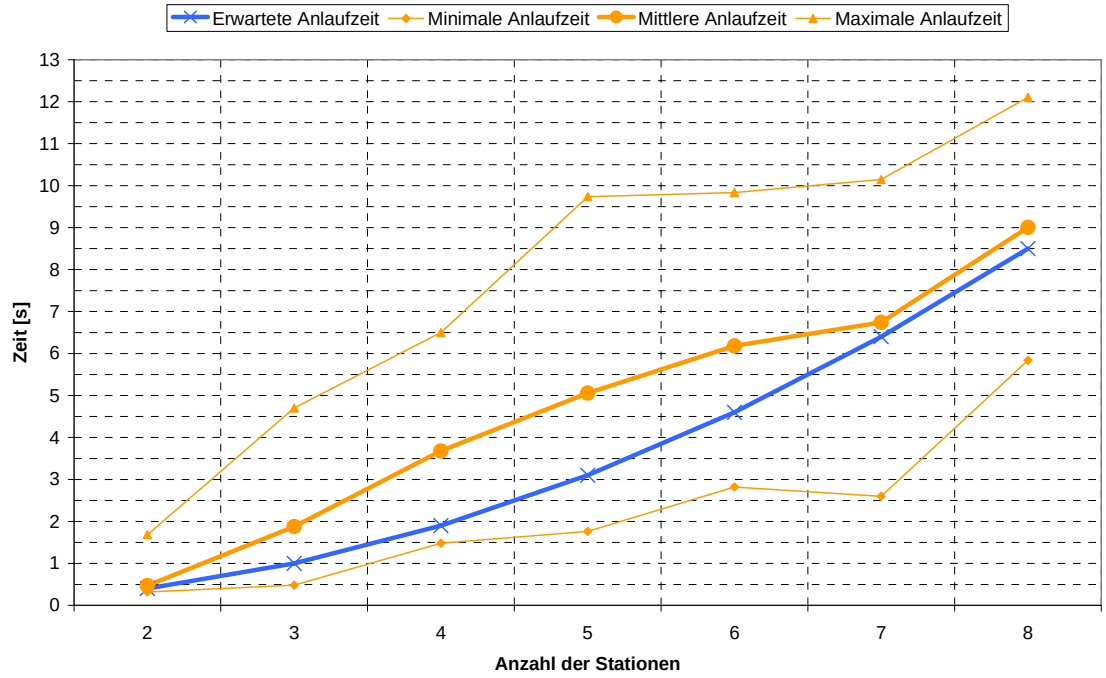

Abbildung 5.5: Initiale Anlaufzeit des IEEE 802.11i bei voller Konnektivität

Typische Topologie

Der Verlauf der Initialen Anlaufzeit des Netzwerks, für die typische Topologie aus Abbildung 5.3, verhält sich, wie in Abbildung 5.6 dargestellt, für die mittlerer Initiale Anlaufzeit weitestgehend linear. Eine Ausnahme stellt das hinzunehmen der Stationen 7 und 8 dar. Hier ist ein schwächerer Anstieg zu verzeichnen, was auf weniger Paketverluste im Vergleich zu den vorangegangenen Messreihen schließen lässt. Eine Ursache hierfür könnte die geringe Konnektivität von nur einer Verbindung zum übrigen Netzwerk sein, so dass im Vergleich zu den vorangegangen Messreihen nur ein geringfügig höherer Kommunikationsaufwand benötigt wird.

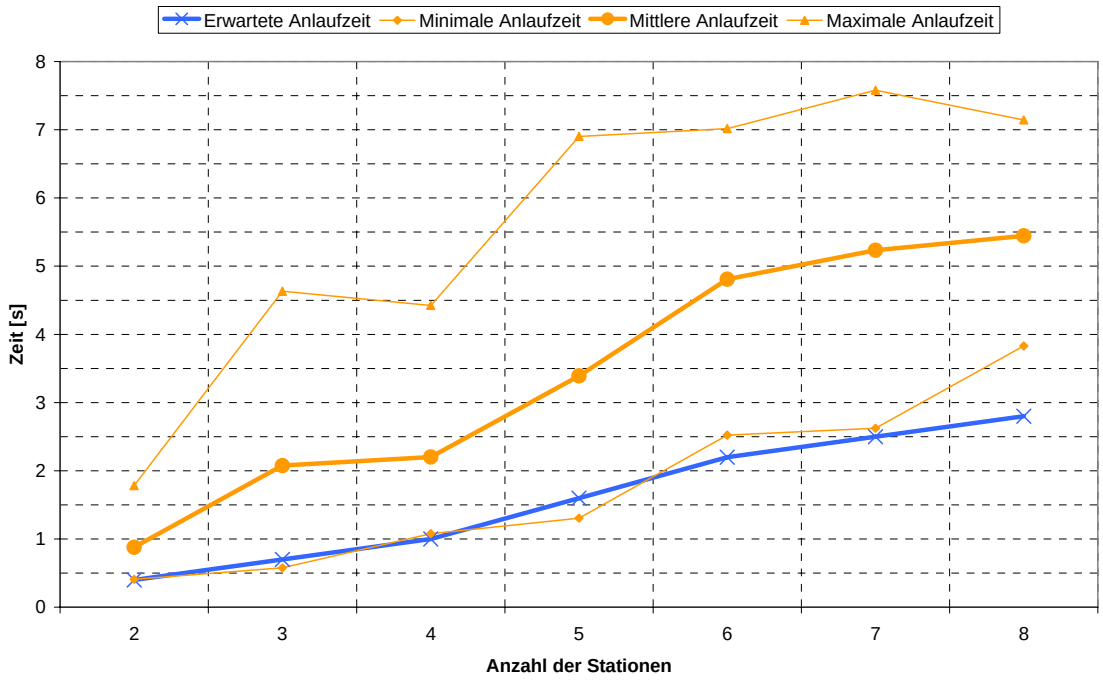

Abbildung 5.6: Initiale Anlaufzeit des IEEE 802.11i in einer typischen Topologie

Die minimale Initiale Anlaufzeit weicht über alle Stationen betrachtet maximal um eine Sekunde ab. Dies bedeutet, dass es in einer Messreihe mindestens eine Messung gab in der keine oder nur wenige Paketverluste aufgetreten sind. Anders sieht es in der mittleren Initialen Anlaufzeit aus. Da ein Paketverlust unter der Berücksichtigung der verwendeten Konfiguration aus Tabelle 5.1 etwa eine Sekunde kostet, zeigt sich, dass im Mittel zwei Paketverluste aufgetreten sind, so dass die mittlere Initiale Anlaufzeit immer oberhalb der erwarteten Initialen Anlaufzeit liegt.

5.1.3 Auswertung – WMN-Modus

Nachdem im vorangegangen Abschnitt bereits die Ergebnisse für den IEEE 802.11i Standard vorgestellt wurden, werden nachfolgend die Ergebnisse für den WMN-Modus präsentiert. Diese wurden unter vergleichbaren Bedingungen ermittelt und werden ebenfalls für die drei verschiedenen Szenarien separat betrachtet.

Ketten-Topologie

In Abbildung 5.7 ist zu sehen, dass die Initialen Anlaufzeiten nahezu linear mit der Anzahl der Stationen zunehmen. Dieser Verlauf entspricht den Erwartungen aus *Abschnitt 5.1.1*, da mit der Hinzunahme der Station 3 und allen nachfolgenden Stationen, die Authentifizierung von der Authentifizierung der jeweiligen Vorgängerstation abhängt. Damit wird zu einem Zeitpunkt auch nur ein Handshake durchgeführt, was das Medium zu einem vernachlässigbaren Maß belastet.

Da alle Werte der mittleren Initialen Anlaufzeit und damit einher auch die der minimalen Initialen Anlaufzeit unterhalb der erwarteten Initialen Anlaufzeit liegen, lässt das den Schluss zu, dass die Authentifizierungs-Anfragen schneller beantwortet wurden als in der Vorbetrachtung erwartet. Dies kann allerdings durch die Messmethode hervorgerufen werden, da zwischen dem Starten der ersten Station und der achten Station bis zu 100 ms vergangen sein können. Damit könnten in dieser Zeit bereits Stationen authentifiziert sein, so dass der Unterschied zwischen zwei Messreihen geringer als erwartet ausfällt.

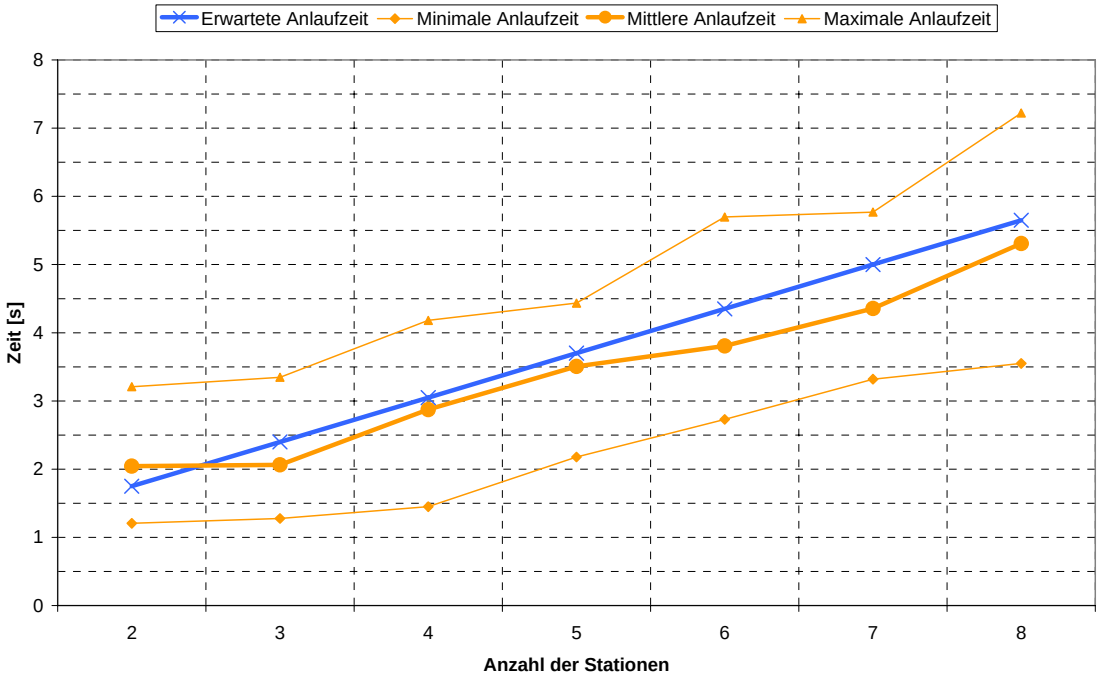

Abbildung 5.7: Initiale Anlaufzeit des WMN-Modus in der Ketten-Topologie

Topologie mit voller Konnektivität

Wie die Abbildung 5.8 zeigt, fällt der Anstieg der Initialen Anlaufzeiten überwiegend gering aus. Dies zeigt, dass auch unter der zunehmenden Anzahl von Client-Stationen nahezu jede Authentifizierung direkt mit der Master-Station stattgefunden hat.

Ein stärkerer linearer Anstieg hingegen, wie er bei der minimalen Initialen Anlaufzeit zwischen fünf und sechs Stationen im Ansatz zu verzeichnen ist, kann bedeuten, dass einige Authentifizierungen mit Stationen vorgenommen wurden, welche sich ihrerseits zuvor authentifizieren mussten. Eine mögliche Ursache hierfür kann in der realen Anordnung der Stationen liegen. So ist es möglich, dass eine Station A eine bessere Konnektivität zu der Master-Station und einer weiteren Station B aufweist, als die Station B direkt zur Master-Station. Damit wäre es möglich, dass Station A bereits authentifiziert ist, während Station B noch keine Antwort auf die Authentifizierungsanfrage erhalten hat, so dass Station A bereits die Authentifizierung von Station B vornehmen kann. Die ermittelten Werte der minimalen Anlaufzeit

weichen um maximal 500 ms und die der mittleren Initialen Anlaufzeit um maximal eine Sekunde ab. Dies könnte durch eine zeitweilige schlechtere Konnektivität oder durch Paketverluste verursacht sein. Damit befinden sich die Messwerte innerhalb der Größenordnung des Erwartungsbildes.

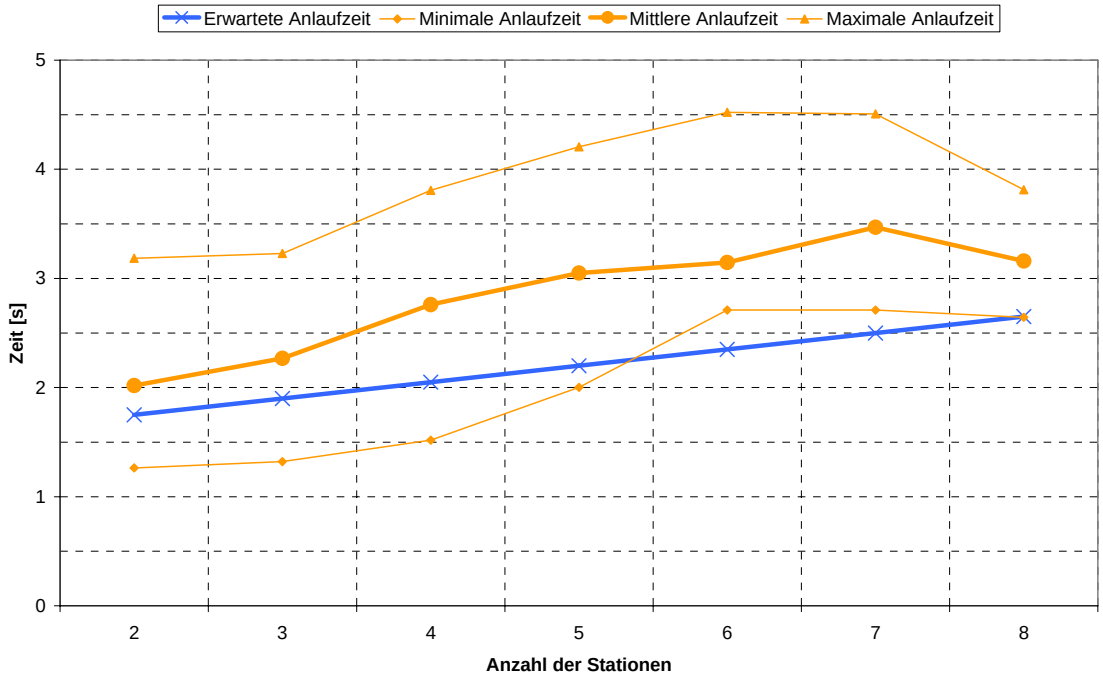

Abbildung 5.8: Initiale Anlaufzeit des WMN-Modus bei voller Konnektivität

Typische Topologie

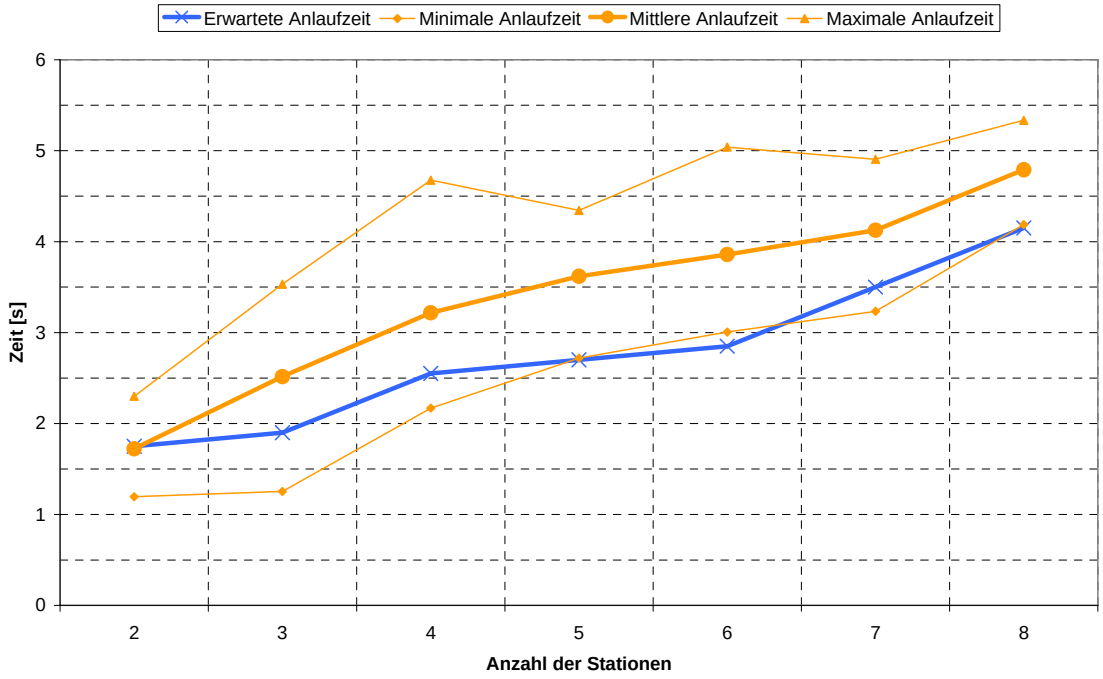

Abbildung 5.9: Initiale Anlaufzeit des WMN-Modus in einer typischen Topologie

Abbildung 5.9 zeigt die Initialen Anlaufzeiten für die in Abbildung 5.3 dargestellte typische Topologie. Auch hier sind einige Teilabschnitte mit geringerem Anstieg gegenüber anderen zu erkennen, was auf Authentifizierung mit der Master-Station hindeutet. Die leicht stärkeren Anstiege gegenüber dem erwarteten Verlauf lassen auf Paktverluste schließen, da lediglich die Stationen 4, 7 und 8 keine Konnektivität zur Master-Station besitzen. Die ermittelten minimalen Initialen Anlaufzeiten bewegen sich mit leichten Abweichungen im erwarteten Bereich. Die mittlere Initiale Anlaufzeit liegt verursacht durch Paketverluste etwas darüber, aber dennoch im erwarteten Bereich.

5.1.4 Vergleich der entwickelten Verfahren

Nachdem in den vorangegangenen beiden Abschnitten die Messergebnisse des IEEE 802.11i Standards und des WMN-Modus separat betrachtet wurden, werden in diesem Abschnitt die Ergebnisse separat für jede Topologie gegenübergestellt und diskutiert.

Ketten-Topologie

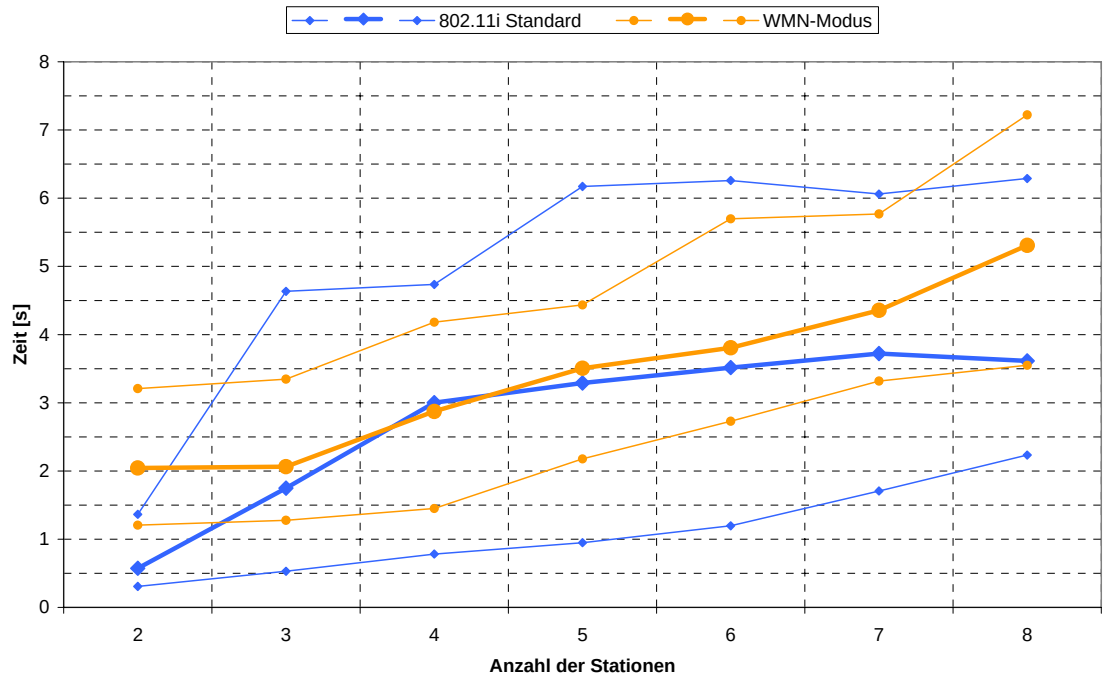

Abbildung 5.10: Vergleich der Initialen Anlaufzeiten in der Ketten-Topologie

Wie in Abbildung 5.10 zu erkennen ist, weißt die mittlere Initiale Anlaufzeit des IEEE 802.11i Standard ab einer Stationsanzahl von vier lediglich einen geringen Anstieg auf, während im WMN-Modus ab einer Stationsanzahl von drei ein stärkerer Anstieg zu erkennen ist. Darüber hinaus ist zu erkennen, dass die mittleren Initialen Anlaufzeiten des IEEE 802.11i Standards, mit Ausnahme einer Topologie von vier Stationen, immer kleiner als die des WMN-Modus sind. Damit zeigt sich, dass der IEEE 802.11i Standard für den Ad-hoc-Modus in der Ketten-Topologie eine, wie aus der Vorbetrachtung in

Abschnitt 5.1.1 bereits zu erwarteten war, kürzere Initiale Anlaufzeit benötigt, was zu einer schnelleren Verfügbarkeit des Netzwerks für die Übertragung von Nutzinformationen führt.

Topologie mit voller Konnektivität

In der Abbildung 5.11 lässt sich an dem Verlauf der mittleren Initialen Anlaufzeit des WMN-Modus erkennen, dass nahezu alle Client-Stationen eine Authentifizierung mit der Master-Station vorgenommen haben, während im Verlauf der mittleren Initialen Anlaufzeit des IEEE 802.11i Standards die deutliche Steigerung des Kommunikationsbedarfes widergespiegelt wird. Es ist deutlich zu erkennen, dass der WMN-Modus bei dieser Topologie, erwartungsgemäß, besser skaliert.

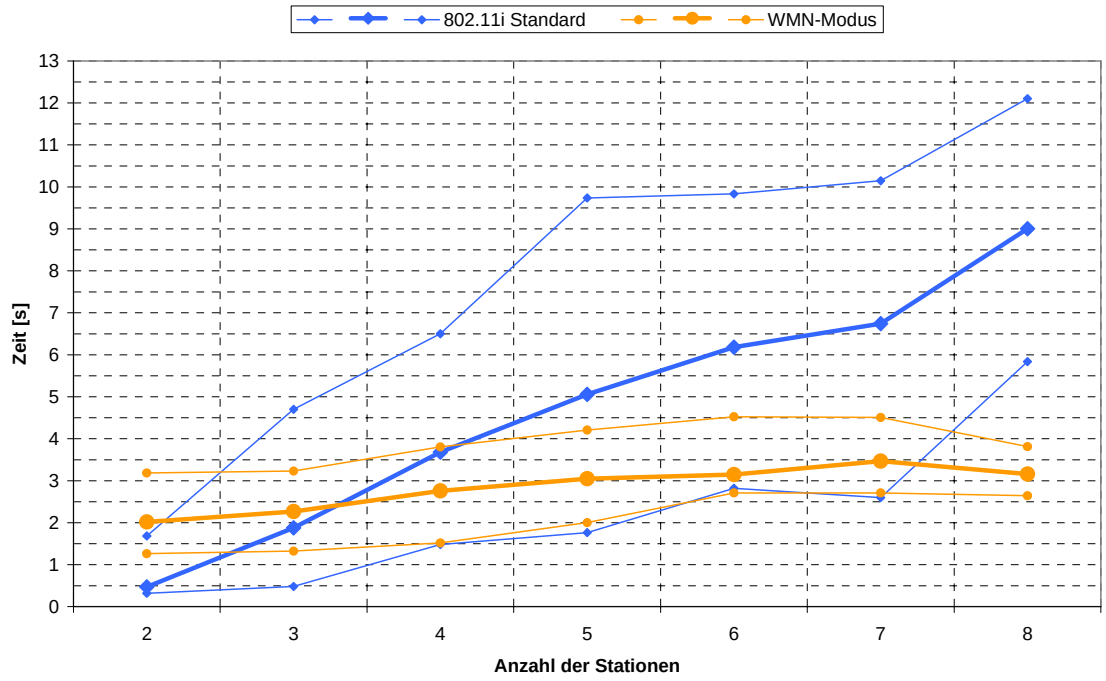

Abbildung 5.11: Vergleich der Initialen Anlaufzeiten bei voller Konnektivität

Typische Topologie

Abbildung 5.12 zeigt für den Verlauf der mittleren Initialen Anlaufzeit für den IEEE 802.11i Standard und den WMN-Modus einen mehrheitlich geringen Anstieg. Wobei der Anstieg des IEEE 802.11i Standards, im Vergleich zum WMN-Modus, in mehreren Teilbereichen stärker ausfällt.

Im Kurvenverlauf ist zu erkennen, dass für eine Anzahl von bis zu fünf Stationen der IEEE 802.11i besser skaliert als der WMN-Modus. Ab einer Stationsanzahl von fünf kehrt sich dies um, wobei im weiteren Verlauf eine Annährung der Initialen Anlaufzeiten zu erkennen ist. Eine Ursache hierfür liegt in der unterschiedlichen Erzeugung der Ereignisse zur Initiierung der Authentifizierung. So wird im IEEE

802.11i Standard die Authentifizierung auf Basis des Beaconings gestartet, während der WMN-Modus eine explizite Authentifizierungsanforderung verlangt. Diese benötigt einen zusätzlichen Zeitaufwand, der sich nur in bestimmten Konstellationen egalisiert bzw. einen Vorteil einbringt.

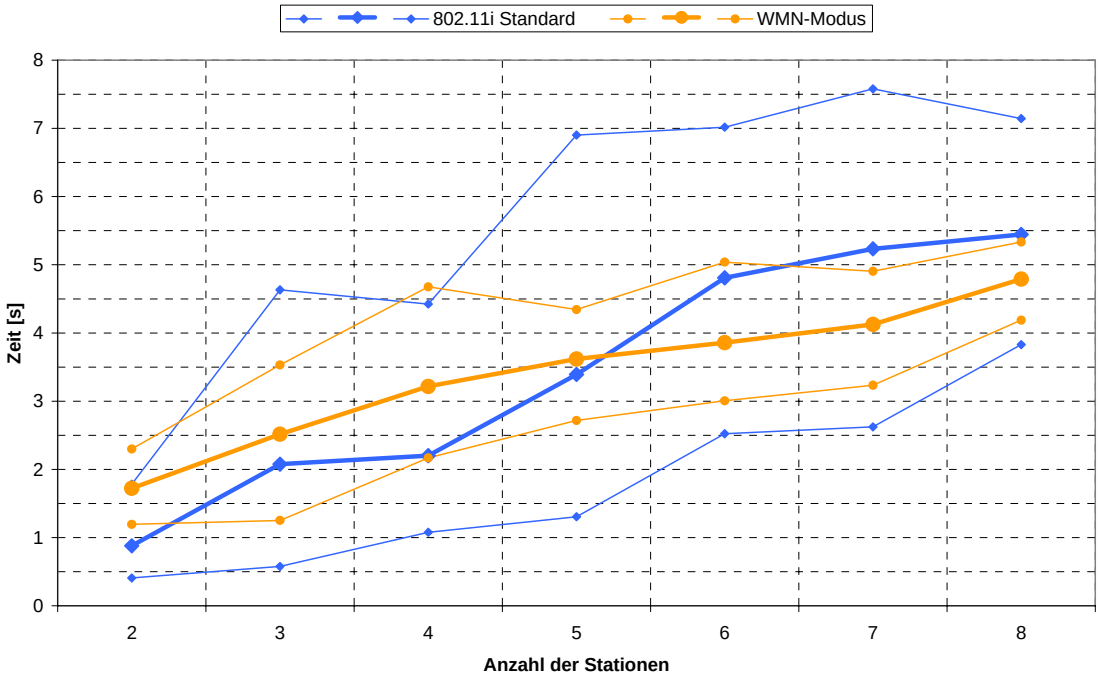

Abbildung 5.12: Vergleich der Initialen Anlaufzeiten einer typischen Topologie

5.2 Re-Authentifizierung im WMN-Modus

5.2.1 Versuchsaufbau

Der WMN-Modus sieht in periodischen Abständen eine Aktualisierung des GKs vor, wobei die Schlüsselaktualisierung mit einer Re-Authentifizierung verbunden ist. Zur Überprüfung der korrekten Funktionsweise der Schlüsselaktualisierung wurde ein Testszenario, bestehend aus drei Stationen, aufgebaut in welchem der Paketverlust während der Übertragung von Nutzdaten ermittelt wurde. Die drei verwendeten Stationen verfügten über uneingeschränkte Konnektivität zueinander und wurden, analog zur Evaluierung in *Abschnitt 5.1.1* mit dem modifizierten MadWifi-Treiber auf Kanal 6 und einer Brutto-Datenrate von 54 MBit/s konfiguriert. Die sich so ergebende Topologie ist in Abbildung 5.13 schematisch dargestellt.

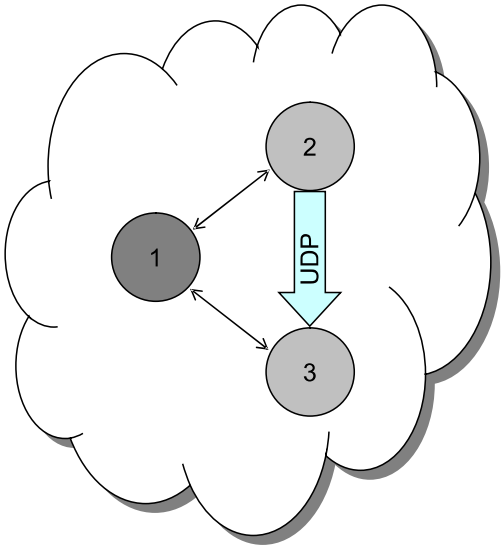

Abbildung 5.13: Topologie zur Bestimmung der Paketverlustes

Um die tatsächlichen Paketverluste zu ermitteln wurde das Netzwerk-Protokoll UDP[1] verwendet, sowie sämtliche Messungen zehnmal nacheinander wiederholt. Als Kenngröße wurde hierzu die von *iperf* ermittelte relative Paketverlustrate verwendet. Diese spiegelt das Verhältnis von Paketverlusten zur tatsächlich übertragenen Paketanzahl wieder, so dass sich neben etwaigen Paketverlusten auf dem Medium, auch mögliche Verluste durch Entschlüsselungsfehler, aufgrund unterschiedlicher Schlüssel beim Verschlüsseln und Entschlüsseln, erkennen lassen. Alle weiteren für das Testszenario relevanten konstanten Konfigurationsparameter sind in der Tabelle 5.8 aufgelistet.

Software	Parameter	Wert
Managementsoftware	Übergangszeit *d*	15 s
Iperf	Laufzeit	180 s
	Datendurchsatz	36 MBit/s
	Größe eines Datagrams	1470 Byte
	UDP Puffer	108 KB

Tabelle 5.8: Parameter zur Bestimmung des Paketverlustes

Anders als TCP[2] verfügt UDP über keine Ratenanpassung, so dass der maximal zu erreichende Datendurchsatz bereits im Vorfeld festgelegt werden muss. So kann eine Sättigung des Mediums erreicht werden, was dazu führt das Paketverluste frühzeitiger auftreten, beispielsweise verursacht durch eine nicht fristgerechte Beendigung einer Re-Authentifizierung. Ob der in UDP vordefinierte Datendurchsatz tatsächlich erreicht wird, hängt jedoch von der maximal erreichbaren Übertragungsgeschwindigkeit ab. Ist diese niedriger als der festgelegte Wert, so wird der maximal erreichbare

[1] User Datagram Protocol
[2] Transmission Control Protocol

Datendurchsatz auf dem Medium erzielt, weil die vom Treiber verwendeten Puffer zum Versenden von Paketen nicht vollständig geleert werden können.

Da der neue Schlüssel bei der Schlüsselaktualisierung nicht sofort für die Verschlüsselung genutzt wird, sondern eine Übergangszeit zur Ermöglichung weiterer Re-Authentifizierungen eingeräumt wird, sollte die Anzahl der Schlüsselaktualisierungen keinen Einfluss auf die Paketverlustrate haben. Auch der Datendurchsatz der Nutzdaten sollte hiervon nicht beeinflusst werden, da die beanspruchte Bandbreite, der mit einer Schlüsselaktualisierung einhergehende Re-Authentifizierung, verhältnismäßig gering ausfällt.

5.2.2 Auswertung

Abbildung 5.14 zeigt den von der Anwendung *iperf* tatsächlich erreichten Datendurchsatz, mit zunehmender Anzahl an Schlüsselaktualisierungen. Es ist zu erkennen, dass der tatsächlich auf dem Medium erzielte Datendurchsatz erwartungsgemäß geringer ausfällt, als die zuvor festgelegten 36 MBit/s und gleichermaßen die Anzahl der Schlüsselaktualisierung keinen Einfluss auf den tatsächlich erzielten Datendurchsatz hat.

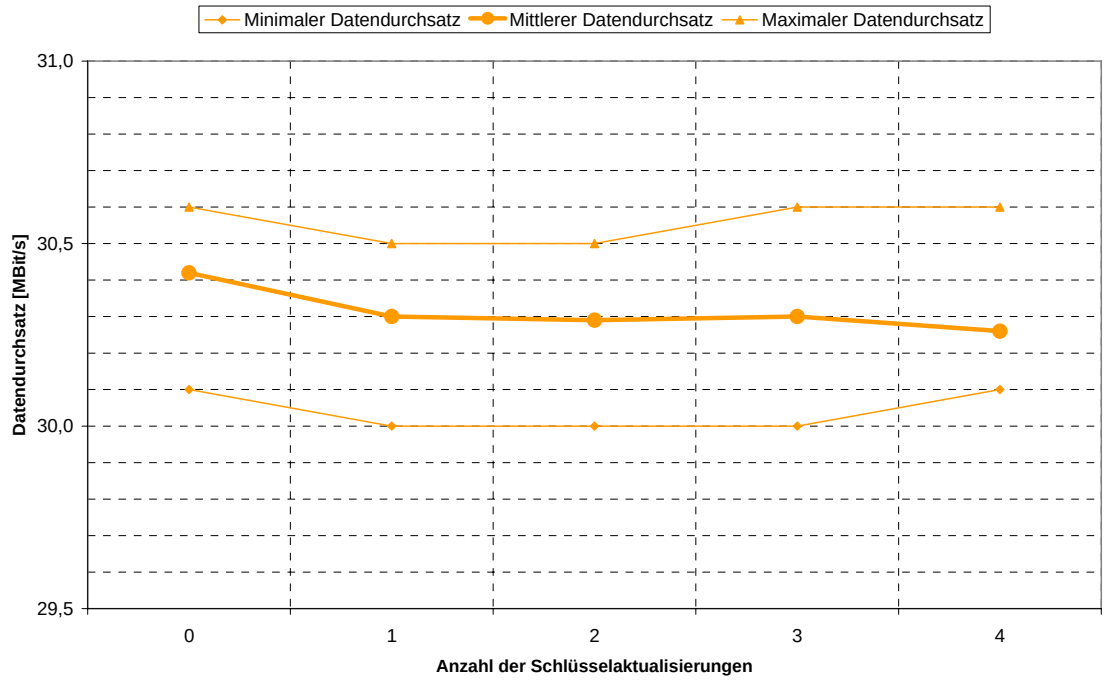

Abbildung 5.14: Erzielter Datendurchsatz der Anwendung

In Abbildung 5.15 ist zu erkennen, dass bei der minimalen und mittleren Paketverlustrate mit zunehmender Anzahl von Schlüsselaktualisierungen die Verlustrate der Pakete in etwa konstant bleibt.

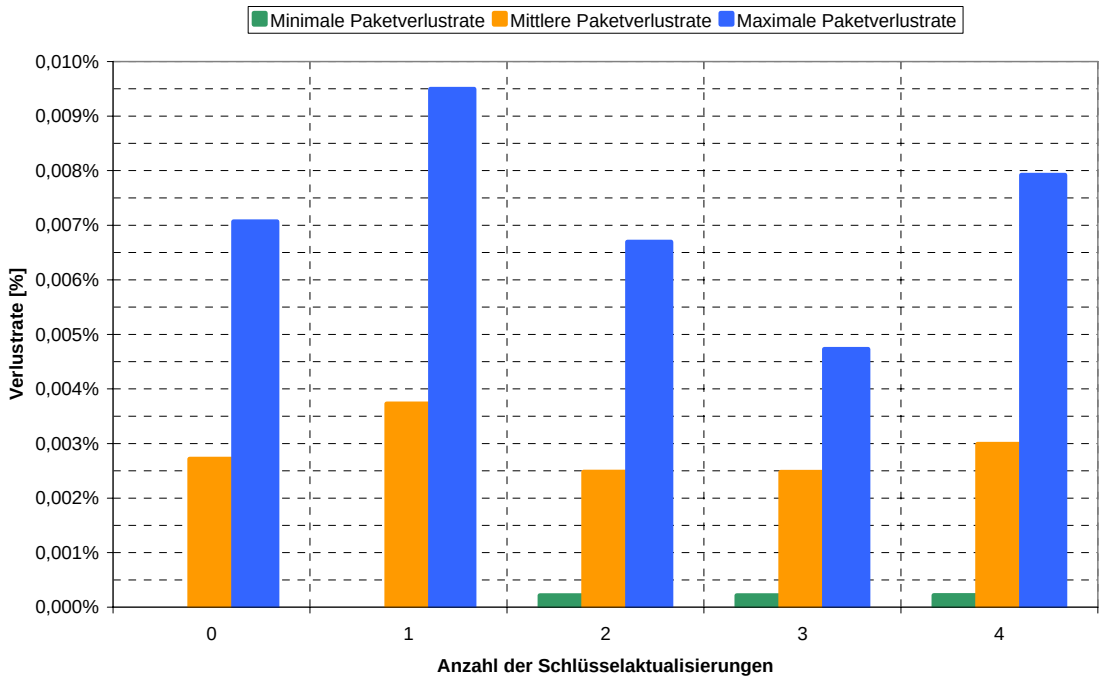

Abbildung 5.15: Verlustrate während der Datenübertragung

Die maximale und mittlere Paketverlustrate bei zwei und drei Schlüsselaktualisierungen liegt sogar unter der Verlustrate ohne Schlüsselaktualisierung. Dies zeigt, dass trotz einer Schlüsselaktualisierung weiterhin eine unterbrechungsfreie Datenübertragung gewährleistet werden kann, da die Häufigkeit der Schlüsselaktualisierung keinen Einfluss auf die relative Paketverlustrate hat. Sowohl die Ergebnisse der Paketverlustrate, als auch der erzielte Datendurchsatz während der Schlüsselaktualisierung entsprechen den, im vorangegangen Abschnitt, dargelegten Erwartungen.

5.3 Ergebnisse

Die vorangegangene Evaluierung hat gezeigt, dass die Implementierung beider Verfahren erwartungsgemäß funktioniert. Es konnte gezeigt werden, dass der Mechanismus der Schlüsselaktualisierung, dem Konzept entsprechend, keinen Einfluss auf die Paketverlustrate und den Datendurchsatz der Nutzdaten hat. Sowohl der IEEE 802.11i Standard, als auch der WMN-Modus konnten die in *Abschnitt 5.1.1* gesteckten Erwartungen, im Hinblick auf die Initiale Anlaufzeit des Netzwerks weitestgehend erfüllen. Die verfahrensbedingte Messungenauigkeit, die auf maximal 100 ms bestimmt werden konnte, stellte sich darüber hinaus bei der Betrachtung der Auswertungen als hinreichend genau heraus.

Neben der Initialen Anlaufzeit des gesamten Netzwerks, lässt sich eine weitere Aussage aus den Untersuchungen ableiten. So benötigt der WMN-Modus eine Authentifizierung um für eine gewisse Zeit zum WMN zu gehören, während der IEEE 802.11i Standard abhängig von den in Reichweite befindlichen Nachbarstationen ist. Damit entspricht die

Initiale Anlaufzeit des IEEE 802.11i Standards für eine Topologie, in etwa der benötigten Zeit zur erneuten Authentifizierung bei selbiger auftretender Topologie in der Nachbarschaft, beispielsweise hervorgerufen durch Mobilität. Der WMN-Modus hingegen, benötigt in einer solchen Situation keine erneute Authentifizierung, womit ein Roaming entfällt und die Flexibilität erhalten bleibt.

6 Zusammenfassung und Ausblick

In diesem letzten Abschnitt werden zunächst die wesentlichen Inhalte der Arbeit zusammengefasst, bevor abschließend ein Ausblick auf mögliche, zukünftige Arbeiten gegeben wird.

6.1 Zusammenfassung

In der vorliegenden Arbeit wurden zwei Security-Verfahren für WMNs entwickelt. Das erste Verfahren wurde auf Grundlage des IEEE 802.11i Standards für Ad-hoc-Netzwerke entwickelt, während das zweite Verfahren eine spezielle Lösung nur für WMNs darstellt.

Im zweiten Abschnitt wurden die Grundlagen zu dieser Arbeit erläutert, sowie bereits vorhandene Vorarbeiten zu dieser Thematik eingearbeitet. Weiterhin wurden die frei verfügbaren Applikationen vorgestellt, auf deren Grundlage die späteren prototypischen Implementierungen der Security-Verfahren für WMNs umgesetzt wurden.

Da der IEEE 802.11i Standard, im Gegensatz zu Infrastruktur-Netzwerken, für Ad-hoc-Netzwerke lediglich Eckpunkte für die Realisierung eines Verfahrens aufschlüsselt, wurde in Abschnitt drei ein Konzept zur Realisierung des IEEE 802.11i Standards für Ad-hoc-Netzwerke erarbeitet. Während dieser Erarbeitung wurde erkannt, dass der IEEE 802.11i Standard für den Ad-hoc-Modus Security-Mechanismen vorsieht, die zu Lasten der Leistungsfähigkeit, speziell bei mobilen Stationen, eines WMNs gehen, jedoch gleichzeitig keine Erhöhung der Security erzielen. Die daraus gewonnenen Erkenntnisse flossen in die Entwicklung eines Konzepts, für ein, auf die Bedürfnisse von WMNs angepasstes Verfahren, ein. So wurde erkannt, dass in WMNs eine Authentifizierung zwischen der nicht authentifizierten Station und einer bereits authentifizierten Station genügt, so dass die Belastung des Mediums, im Vergleich zum IEEE 802.11i Standard für den Ad-hoc-Modus, sinkt und die Flexibilität, insbesondere für mobile Stationen, steigt. Darüber hinaus konnte ebenfalls der Aufwand des Schlüsselmanagements reduziert werden, indem lediglich ein Schlüssel für alle Stationen des WMNs verwendet wird. Dieser wird zudem automatisch in bestimmten Zeitintervallen aktualisiert.

Der vierte Abschnitt legte Schlüsselstellen bei der Implementierung der entwickelten Konzepte dar. Bei der prototypischen Umsetzung des IEEE 802.11i Standards ergab sich allerdings ein Problem in der Implementierung, der im Konzept entwickelten Lösung zur Verwaltung von GTKs als Empfängerschlüssel, in MadWifi. Dieses Problem stellt kein konzeptionelles Problem dar, konnte aber dennoch im zeitlichen Rahmen dieser Arbeit nicht zufriedenstellend gelöst werden, so dass in der prototypischen Implementierung eine Sicherung von Multicast-Verbindungen, lediglich für drei Nachbarstationen möglich ist.

In Abschnitt fünf wurden schließlich die entwickelten Verfahren auf ihre Funktionalität und ihre Leistungsfähigkeit, anhand von verschiedenen Testszenarien, untersucht. Dabei konnte gezeigt werden, dass beide prototypischen Implementierungen erwartungsgemäß funktionstüchtig sind und im Wesentlichen die erwartete Performance an den Tag legen. Jedoch hat sich aus Sicht der Flexibilität in Kombination mit der erzielten Leistungsfähigkeit, sowie aufgrund der eingeschränkten Möglichkeit der Verwendung von Gruppenschlüsseln in der prototypischen Umsetzung des IEEE 802.11i Standards für Ad-hoc-Netzwerke, nur die prototypische Realisierung des WMN-Modus für die praktische Verwendung in WMNs empfohlen.

6.2 Ausblick

Im Rahmen der Arbeit haben sich bei der Implementierung der entwickelten Verfahren einige Probleme ergeben, die innerhalb dieser Arbeit nicht gelöst werden konnten. So ist es nicht gelungen den Schlüsselcache von MadWifi zur Verwaltung von GTKs als Empfängerschlüssel zu verwenden. Auch konnte der konzipierte Mechanismus des WMN-Modus gegen Replay-Angriffe nicht in MadWifi integriert werden, so dass die prototypische Implementierung derzeit keinen aktiven Schutz gegen Replay-Angriffe aufweist. Auch die Erweiterung des WMN-Modus, um Mechanismen der Leader Election oder um Mechanismen für die verteilte Erzeugung des GKs, würde den WMN-Modus weiter aufwerten.

Zur leichteren Umsetzung einer Zugangskontrolle, könnten zudem beide Verfahren mit Upper-Layer Authentifizierungsmethoden an Authentifizierungs-Server, im Backbone, angebunden werden. Hierzu müsste ein entsprechendes Konzept erarbeitet werden und, möglicherweise, unter Verwendung der bereits vorgesehenen Schnittstellen, realisiert werden. Dies würde die manuelle Konfiguration eines PSK erübrigen, was bei großen WMNs einen nicht unerheblichen Zeitaufwand beansprucht.

Da der WMN-Modus lediglich einen GK für die Verschlüsselung der Daten verwendet, wird dieser, im Vergleich zu der Schlüsselnutzung im IEEE 802.11i Standard für Ad-hoc-Netzwerke, häufiger verwendet. Dies könnte eine Abschwächung der Security bedeuten, die es noch zu untersuchen gilt.

Auch der in *Abschnitt 2.3.5* vorgestellt IEEE P802.11s enthält ein Security-Konzept. Da dieses zum derzeitigen Zeitpunkt noch unvollständig ist, könnte es zukünftig, nach dessen Vervollständigung, umgesetzt werden und so mit den beiden, in dieser Arbeit entwickelten Verfahren, hinsichtlich seiner Leistungsfähigkeit in WMNs untersucht werden.

Literaturverzeichnis

[AWDS08] Ad-hoc Wireless Distribution System: *http://awds.berlios.de/*. 26. Mai 2008.

[AWW04] Ian F. Akyildiz, Xudong Wang, Weilin Wang: *Wireless mesh networks: a survey*. Atlanta, 2004.

[BGW01] Nikita Borisov, Ian Goldberg, David Wagner: *Intercepting Mobile Communications: The Insecurity of 802.11*. Berkeley, 2001.

[CG97] Pau-Chen Cheng, Robert Glenn: *Test Cases for HMAC-MD5 and HMAC-SHA-1*. Yorktown Heights, 1997.

[CRK05] Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman: *Linux Device Drivers*. Stanford, 3rd Edition, 2005.

[DaRi99] Joan Daemen, Vincent Rijmen: *AES Proposal: Rijndael*. Brüssel, 1999.

[DiHe76] Whitfield Diffie, Martin E. Hellman: *New Directions in Cryptography*. Stanford, 1976.

[Du08] Jon Dugan u.w.: *http://sourceforge.net/projects/iperf/*. 14. August 2008.

[Eck04] Claudia Eckert: *IT-Sicherheit: Konzept – Verfahren – Protokolle.* Oldenbourg, München, 3.Auflage, 2004.

[EdAr04] Jon Edney, William A. Arbaugh: *Real 802.11 Security: Wi-Fi Protected Access and 802.11i.* Addison-Wesley, Boston, 2004.

[FG97] Chane L. Fullmer, J.J. Garcia-Luna-Aceves: *Solutions to Hidden Terminal Problems in Wireless Networks*. Santa Cruz, 1997.

[FMS01] Scott Fluhrer, Itsik Mantin, Adi Shamir: *Weaknesses in the Key Scheduling Algorithm of RC4*. San Jose, 2001.

[HPS99] Kostas P. Hatzis, George P. Pentaris, Paul G. Spirakis, Vasilis T. Tampakas, Richard B. Tan: *Fundamental Control Algorithms in Mobile Networks*. Patras, 1999.

[IEE99] IEEE: *IEEE 802.11 Standard - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, 1999.

[IEE07] IEEE: *IEEE 802.11 Standard - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, 2007.

[IEE99a] IEEE: *IEEE 802.11a Standard - High-speed Physical Layer in the 5 GHz Band*. New York, 1999.

[IEE03g] IEEE: *IEEE 802.11g Standard - Further Higher Data Rate Extension in the 2.4 GHz Band*. New York, 2003.

[IEE03h] IEEE: *IEEE 802.11h Standard - Spectrum and Transmit Power Management Extensions in the 5 GHz band in Europe*. New York, 2003.

[IEE04] IEEE: *IEEE 802.11i Standard - Medium Access Control (MAC) Security Enhancements*. New York, 2004.

[IEE08] IEEE: *IEEE P802.11s / D2.0 - Mesh Networking*. New York, 2008.

[IEE04X] IEEE: *IEEE 802.1X Standard - Port-Based Network Access Control*. New York, 2004.

[KBC97] Hugo Krawczyk, Mihir Bellare, Ran Canetti: *HMAC: Keyed-Hashing for Message Authentication*. Yorktown Heights, 1997.

[Kin01] Jason S. King: *An IEEE 802.11 Wireless LAN Security White Paper*. Livermore, 2001.

[Kob87] Neal Koblitz: *Elliptic Curve Cryptosystems*. Seattle, 1987.

[KR05] Ted Krovetz, Phillip Rogaway: *The OCB Authenticated-Encryption Algorithm*. Davis, 2005.

[MaH08] Jouni Malinen: *http://hostap.epitest.fi/hostapd/*. 16. April 2008.

[MaS08] Jouni Malinen: *http://hostap.epitest.fi/wpa_supplicant/*. 16. April 2008.

[MW08] MadWifi.org: *http://madwifi.org/*. 16. April 2008.

[MWV00] Navneet Malpani, Jennifer L. Welch, Nitin Vaidya: *Leader Election Algorithms for Mobile Ad Hoc Networks*. College Station, 2000.

[NIST01] National Institute of Standards and Technology: *Announcing the ADVANCED ENCRYPTION STANDARD (AES)*. Gaithersburg, 2001.

[NIST02] National Institute of Standards and Technology: *Announcing the SECURE HASH STANDARD (FIPS PUB 180-2)*. Gaithersburg, 2002.

[NMG01] Edgar Nett, Michael Mock, Martin Gergeleit: *Das drahtlose Ethernet – Der IEEE 802.11 Standard: Grundlagen & Anwendung*. Addison-Wesley, München, 2001.

[NWG02] Network Working Group: *Advanced Encryption Standard (AES) Key Wrap Algorithm*. Herndon, 2002.

[NWG04] Network Working Group: *Extensible Authentication Protocol (EAP)*. Redmond, 2004.

[NWG05] Network Working Group: *The Kerberos Network Authentication Service*. Marina del Rey, 2005.

[NWG06] Network Working Group: *The Transport Layer Security (TLS) Protocol*. Palo Alto, 2006.

[RBBK01] Phillip Rogaway, Mihir Bellare, John Black, Ted Krovetz: *OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption*. Davis, 2001.

[RWSX07] Andreas Roos, Sabine Wieland, Andreas Th. Schwarzbacher, Bangnan Xu: *Time behaviour and network encumbrance due to authentication in wireless mesh access networks*. Dublin, 2007.

[Riv92] Ronald L. Rivest: *The MD5 Message-Digest Algorithm*. Cambridge, 1992.

[Rog01] Phillip Rogaway: *PMAC - Proposal to National Institute of Standards and Technology (NIST) for a parallelizable message authentication code*. Davis, 2001.

[Sch96] Bruce Schneier: *Angewandte Kryptographie - Protokolle, Algorithmen und Sourcecode in C*. Addison-Wesley, Bonn, 1996.

[Sch00] Thomas Schmidt: *CRC Generating and Checking*. Atlanta, 2000.

[Tan03] Andrew S. Tanenbaum: *Moderne Betriebssysteme*. Pearson Studium, München, 2.Auflage, 2003.

[Tsa04] Chii-Ren Tsai: *Non-Repudiation In Practice*. Silver Spring, 2004.

[TWP07] Erik Tews, Ralf-Philipp Weinmann, Andrei Pyshkin: *Breaking 104 bit WEP in less than 60 seconds*. Darmstadt, 2007.

[Wal02a] Jesse Walker: *802.11 Key Management Series - Part I: Key Management for WEP and TKIP*. Santa Clara, 2002.

[Wal02b] Jesse Walker: *802.11 Security Series - Part II: The Temporal Key Integrity Protocol (TKIP).* Santa Clara, 2002.

[WiFi03] Wi-Fi Alliance: *Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks.* Austin, 2003.

[WiFi05] Wi-Fi Alliance: *Deploying Wi-Fi Protected Access (WPA™) and WPA2™ in the Enterprise*. Austin, 2005.

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit selbständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Magdeburg, den 16. September 2008

_____

Christian Fackroth