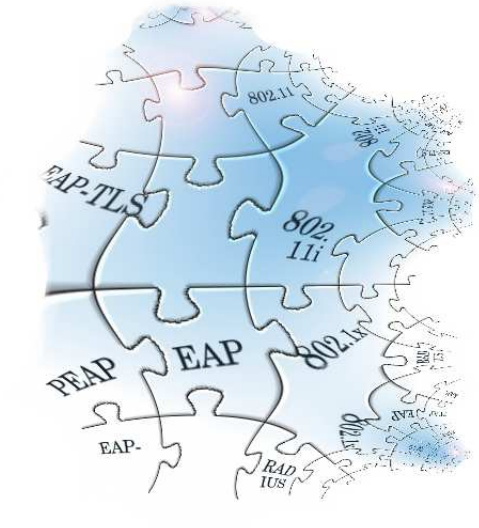
Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund

Studienarbeit

Netzwerkauthentifizierung im WLAN



von
Thomas Otto

Aufgabenstellung und Betreuung:
Prof. Dr. Stefan Fischer
Dipl.-Wirt.-Inf. Stefan Schmidt

Braunschweig, 27. April 2004

Erklärung

Ich versichere, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Braunschweig, 27. April 2004 Thomas Otto

Übersicht

Der neue WLAN Standard IEEE 802.11i bringt verbesserte Sicherheitsmechanismen hinsichtlich Authentifizierung, Zugangskontrolle und Vertraulichkeit mit. In dieser Arbeit soll die Authentifizierung genauer betrachtet werden, die auf dem Standard IEEE 802.1x basiert. Zum Transport von Nachrichten zwischen den verschiedenen beteiligten Einheiten ist die Wahl auf das Extensible Authentication Protocol (EAP) gefallen. Es ist unabhängig von der zugrundeliegenden Sicherungsschicht und kann neben PPP oder 802.3 Ethernet auch über 802.11 WLAN laufen. Sowohl 802.1x als auch EAP sind nur ein Rahmenwerk. Die Authentifizierung erfolgt durch konkrete EAP Methoden.
Die Absicht dieser Arbeit ist es zum einen, das mitunter komplexe Zusammenspiel von 802.1x, EAP und RADIUS im Zusammenhang der WLAN Authentifizierung aufzuzeigen. Zum anderen werden mehrere aktuelle EAP Methoden präsentiert und abgewogen, ob sie im Wireless Umfeld einsetzbar sind. Hierzu ist zunächst auf die Gefahren, denen eine EAP Methode ausgesetzt ist, und die daraus resultierenden Anforderungen an sie einzugehen. Statt aber jeweils in einer Protokollanalyse zu münden, wird vielmehr das Ziel verfolgt, Schlüsseltechniken und -mechanismen herauszuarbeiten und allgemeine Designziele vorzustellen. Die vorgestellten Verfahren sind - in dieser Reihenfolge - EAP-MD5, LEAP, EAP-TLS, PEAPv2, EAP-TTLS, EAP-FAST, EAP-SIM/AKA und EAP-SPEKE.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

"Ich bin drin" - dieser Werbeslogan eines Internetdienstanbieters hat uns gut ein Jahr begleitet. Online zu sein, den Zugang zur neuen Welt, dem Internet, zu haben und dieses intensiv zu nutzen ist aus heutiger Sicht nicht mehr wegzudenken. Dabei an einen festen Platz gebunden zu sein, wird zukünftig auch nicht mehr Voraussetzung sein, denn Portabilität bis Mobilität und Konnektivität gehen als Fortschritte in diese Entwicklung mit ein, so dass der Spruch zu "Ich bin immer drin" im Sinne von jederzeit und überall angepasst werden muss.
Unser heutiges Leben zeichnet sich durchweg durch Mobilität ab. Nicht nur im Telefoniebereich, sondern auch im Netzwerkbereich hält dieser Trend Einzug, denn drahtlose Netzwerke erfreuen sich immer größerer Beliebtheit und werden in immer zunehmendem Maße eingesetzt. Die Vorteile im Vergleich zu herkömmlichen, d.h. auf Kabeln basierenden Netzwerken sind nicht von der Hand zu weisen. Stetig fallende Anschaffungskosten der Hardware, annähernd so hohe Datenübertragungsraten, eine äußerst schnelle Inbetriebnahme und geringe Installationskosten durch Wegfallen des Kabelverlegens sind Beispiele hierfür.
Natürlich gibt es in dieser Welt auch Schattenseiten - Netzwerke sind seit jeher Gegenstand von Angriffen und Attacken, angetrieben von vielfältigsten Motiven. Sicherheit in Netzwerken ist bereits ein äußerst komplexes und schwieriges Thema. Netzwerksicherheit in drahtlosen Netzen ist aber eine unvergleichbare Herausforderung. Aufgrund der physikalischen Eigenschaften der Luftschnittstelle lassen sich die Daten auch weit außerhalb des gewünschten Einsatzgebietes empfangen und entsprechend Daten einspielen. Der Angreifer kann aus großer Distanz operieren, ohne dass herkömmliche physische Zugangskontrollen ansetzen könnten. Zusammenfassend ergeben sich eine Vielzahl weiterer Angriffspunkte, mit denen man sich auseinandersetzen und Gegenmaßnahmen entwickeln muss.
Seit Verabschiedung des 802.11 WLAN Standards im Jahre 1997 und Veröffentlichung in 1999 sind auf technischer Seite fortlaufend Verbesserungen erzielt worden, indem die anfängliche Datendurchsatzrate von 1 bis 2 MBit/s über 802.11b mit 11 MBit/s bis schließlich 54 MBit/s in 802.11g gesteigert werden konnte. Letzterer wurde übrigens erst im Juni 2003 standardisiert. Unverändert seit Bestehen von 802.11 blieben hingegen die Sicherheitsmechanismen. In den Jahren 2000 und 2001 wurden sie infolge von Untersuchen in ihrer Gesamtheit als angreifbar, die meisten als vollkommen wirkungslos, aufgedeckt.
Als Gegenmaßnahme hat die IEEE eine Arbeitsgruppe "i" ins Leben gerufen, die mit der Aufgabe betraut wurde, die Sicherheitsarchitektur von 802.11 vollkommen zu überarbeiten. Die Ergebnisse der über drei Jahre andauernden Anstrengungen werden im Standard 802.11i festgehalten. Es werden Möglichkeiten der Authentifizierung, Authorisierung und besseren Verschlüsselung definiert, wobei diese Arbeit dem ersten Punkt gewidmet ist. Dabei nimmt die Authentifizierung eine Schlüsselrolle ein. Nur wenn sie erfolgreich verlaufen ist, erhält der Benutzer die Berechtigung zur Teilnahme an der Kommunikation im WLAN.

Zur Terminologie

Da die Arbeit in deutscher Sprache verfasst ist, in der Computerwelt aber einheitlich Englisch verwendet wird, ist an mehreren Stellen zwischen Beibehalten des Originalausdrucks und Übersetzen zu entscheiden gewesen. Sofern es inhaltlich vertretbar und sprachlich möglich war, ist eine Übersetzung erfolgt. Drei Begriffe sollen den Abwägeprozess illustrieren, und zwar "mutual authentication", "ciphersuite", sowie "Master Session Key". Der erste Ausdruck kann unproblematisch mit "gegenseitige Authentifizierung" übersetzt werden. Mit "Ciphersuite" muss anders verfahren werden, da die deutsche Sprache auch nicht annähernd ein Äquivalent beinhaltet. Versuche wie Chiffresammlung, Verschlüsselungsfolge oder sogar Zifferngarnitur erweisen sich als ungenau oder unsinnig. Manche Bezeichnungen wie der "Master Session Key" oder das TLS "Master Secret" sollten, um nicht Verwirrung zu stiften, sogar in der Originalsprache verbleiben.
Die unterschiedliche Schreibweise von Maßangaben (40bit, 40 Byte) ist bewusst erfolgt und soll einer Verwechslung beim Lesen zuvorkommen. Werden Zeichenketten aneinandergehängt, also konkateniert, so wird dies durch | statt durch + ausgedrückt. Insbesondere, wenn die Zeichenketten aus Ziffern bestehen, soll der Fehldeutung einer Addition zuvorgekommen werden. In der Regel wird bei Fachausdrücken aus dem Englischen auch auf eine strenge Deklination verzichtet, was sich vor allem an Durchsetzen des Genitivs richtet. Was Groß- und Kleinschreibung betrifft, so werden Bezeichner in Formeln oder Nachrichten in Protokollen in der Regel unverändert gelassen, bei EAP-TLS beispielsweise "master secret" oder "client_hello". Im Fließtext wird die Kleinschreibung zugunsten der Lesbarkeit aufgegeben, was in diesem Fall in "Master Secret" und "Client_Hello" resultiert. Zudem wird aus dem Kontext deutlich, es nicht mit zwei verschiedenen Nachrichten o.ä. zu tun zu haben.

Kapitel 2

Grundlagen

2.1 Sicherheit in Informationssystemen

In diesem Zusammenhang ist zwischen Sicherheitszielsetzungen, Sicherheitsdiensten und Sicherheitsmechanismen zu unterscheiden. Als grundlegende Zielsetzungen der Informationssicherheit (security objectives) sind Vertraulichkeit (confidentiality), Integrität (integrity), Verfügbarkeit (availability) und legitime Nutzung anzusehen, wobei die ersten drei unter dem CIA-Modell weitläufig bekannt sind. Diese Zielsetzungen werden durch fünf entsprechende Sicherheitsdienste (security services) erreicht, wie sie zum Beispiel der ISO Standard 7498-2 festlegt.

Authentifizierung Der Teilnehmer, ob Gerät oder Benutzer, hat einen Nachweis seiner angegebenen, behaupteten Identität zu erbringen. In der Regel bedingt die Authentifizierung eine Authorisierung des Teilnehmers. Der Authentifizierungsdienst existiert in 2 Varianten. Wird die Identität bei dem Zugriff auf eine Ressource verifiziert, spricht man von Partner-Authentifizierung (peer authentication). Soll dagegen die Herkunft einer Nachricht zweifelsfrei bestimmt werden, muss die Nachricht von einer zu verifizierenden Identität begleitet werden. Der entsprechende Sicherheitsdienst wird Authentifizierung der Datenherkunft (data origin authentication) genannt.

Zugriffskontrolle (access control) stellt sicher, dass ein Zugang zu und die Nutzung von bestimmten Ressourcen des Systems nur authorisierten Teilnehmern und nur in authorisierter Art und Weise möglich ist.

Vertraulichkeit Mit dem Vertraulichkeitsdienst soll der Schutz der Privatsphäre umgesetzt werden. Vertraulichkeit ist der Schutz von Informationen im System, derart, dass nicht-authorisierte Teilnehmer auf diese auch nicht zugreifen können.

Datenintegrität bezieht sich auf Fehlerfreiheit, Konsistenz und Vollständigkeit von Daten. Dieser Sicherheitsdienst umfasst den Schutz der Daten vor jeglicher Art von nicht-authorisierter Veränderung, d.h. Einfügen, Löschen, Umstellen, Wiedereinspielen und inhaltliche Manipulation.

Nichtabstreitbarkeit (non-repudiation; oder auch "Nachweisbarkeit"). Dem Absender bzw. dem Empfänger wird ein Beweis über die Zustellung der Informationen erbracht. Ziel dieses Sicherheitsdienstes ist es, ein fälschliches nachträgliches Abstreiten der stattgefundenen Kommunikation durch eine der Kommunikationsparteien zu verhindern.

Diese fünf Sicherheitsdienste werden durch Anwendung geeigneter Sicherheitsmechanismen erreicht. Die wichtigsten angewandten Sicherheitsmechanismen sind Verschlüsselung,

digitale Signaturen, Zugriffskontrolllisten (ACLs), Authentifizierungsvorgänge, Routingkontrolle oder Beglaubigungen durch Dritte. Dabei können sich durchaus mehrere Mechanismen für einen Sicherheitsdienst anbieten oder ein Mechanismus mehrere Sicherheitsdienste erfüllen.

Da sich diese Arbeit mit Authentifizierung beschäftigt, ist dieser Aspekt genauer zu betrachten. Die Authentifizierung in WLAN 802.11i dient dazu, die Berechtigung für den Zugriff auf geschützte Ressourcen oder Dienste zu erhalten bzw. diese nutzen zu können. Der Nachweis kann durch ein der Identität eindeutig zugeordnetes

- Wissen (z.B. Passwort)
- Besitz (Chipkarte, Schlüssel, Ausweis mit Magnetstreifen, digitales Zertifikat)
- Eigenschaft (biometrisches Merkmal wie Fingerabdruck, Iris des Auges)

erfolgen. Ein solcher Berechtigungsnachweis wird als Credential bezeichnet. Die von 802.11i verwendeten Authentifizierungsverfahren (EAP Methoden) basieren auf Credentials wie der Benutzername / Passwort Kombination, digitalen Zertifikaten oder Hardware Token (wie Smartcards, GSM SIM Karte oder UMTS USIM Karte).

Diese Sicherheitsmechanismen in ihrer zugedachten Aufgabe außer Kraft zu setzen ist das Ziel eines Angreifers. Der neue WLAN Standard sieht seine Aufgabe in der Definition von Sicherheitsmechanismen, um die oben genannten Sicherheitsdienste umzusetzen. Als Voraussetzung für ein Mitwirken im Netzwerk und das Nutzen der angebotenen Dienste hat eine Authentifizierung des Teilnehmers stattzufinden, so dass unauthorisierte Geräte wie Benutzer bereits am "Rand des Netzwerks" erkannt werden und keinen Zugang zum Netzwerk erlangen. Diese Zugangskontrolle kann auch für die Vergabe von Zugriffsrechten dienen. Durch Einführung stärkerer kryptographischer Methoden soll die Vertraulichkeit und Datenintegrität sichergestellt werden.

2.2 WLAN 802.11 Betriebsmodi

Der 802.11 Standard definiert zwei Betriebsarten von WLANs, den Infrastruktur Modus und den Ad-Hoc Modus (Abbildung 2.1). Im Infrastruktur Modus kommunizieren die Clients über eine spezielle Basisstation, den Access Point, miteinander. Dieser kann, sofern gewünscht, auch als Brücke zwischen dem drahtlosen und dem drahtgebundenen Netz dienen, indem er Pakete zwischen den Netzen vermittelt.

In der zweiten Betriebsart ist diese Einheit ausgelassen, so dass Clients durch Aufbau von Punkt-zu-Punkt Verbindungen direkt miteinander kommunizieren. Für den Rest der Arbeit soll der unkontrollierte Ad-Hoc Modus von der Betrachtung ausbleiben und ausschließlich der Infrastruktur Modus unterstellt werden, da nur in diesem Rahmen eine sinnvolle Authentifizierung erfolgen kann. Eine Einheit, ob der Access Point selbst oder eine dritte dahinterliegende, entscheidet über den Zugang eines Clients, indem sie im Besitz aller Credentials ist bzw. auf sie Zugriff hat. Mit dieser Einheit findet auch die Authentifizierungsverhandlung statt.

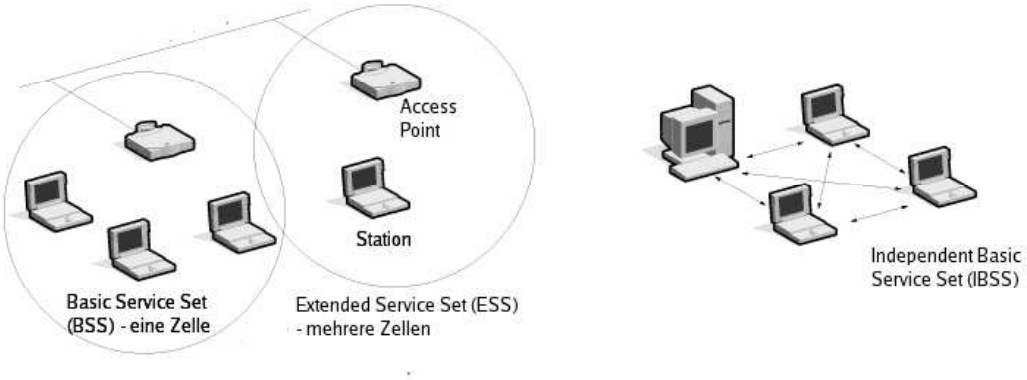


Abbildung 2.1: Infrastruktur Modus, Ad-Hoc Modus

2.3 Sicherheit in WLAN 802.11

Die Bereiche Vertraulichkeit, Datenintegrität und Authentifizierung werden von WEP (Wired Equivalent Privacy) abgedeckt. WEP ist auf der Sicherungsschicht angesiedelt und hat den Schutz der Daten, die im BSS zwischen Client und Access Point bzw. im IBSS zwischen Client und Client transferiert werden, sicherzustellen. Der Schutz reicht wohlgemerkt aber nicht von Ende zu Ende, sondern umfasst nur den drahtlosen Teil des Netzwerks.

Der IEEE Standard [802.11-99] sieht die Aufgabe von WEP darin, authorisierte Benutzer eines WLANs vor gewöhnlichem Mitlauschen zu schützen:

> "The service is intended to provide functionality for the wireless LAN equivalent to that provided by the physical security attributes inherent to a wired medium".

Im Zeitraum Oktober 2000 bis August 2001 (vgl. [ArWi]) wurden in WEP zahlreiche Schwächen identifiziert, die es letzlich praktisch wirkungslos werden ließen. Es kommt weder seinem ursprünglichen Designziel von wired-equivalent Vertraulichkeit noch von Integrität und Authentifizierung nach. Anders ausgedrückt sind alle in [802.11-99] definierten Sicherheitsmechanismen als nutzlos herausgestellt worden. Dabei wirken sich zwei generelle Einschränkungen ([Cw03]) bereits als äußerst nachteilig aus.

- Die Verwendung von WEP ist optional ([802.11-99], 8.2.2) und wird infolgedessen in vielen WLAN Installationen gar nicht erst aktiviert.
- WEP benutzt einen einzigen von allen teilnehmenden Geräten geteilten Schlüssel, der in der Regel auf jedem Gerät gespeichert ist. Ein Verlust eines Gerätes kompromittiert das gesamte Netzwerk. Da der Standard kein Verfahren zur Schlüsselverteilung vorgibt, erfolgt diese in der Regel manuell, so dass eine Verteilung eines neuen Schlüssels mit erheblichem Aufwand verbunden ist.

2.4 Funktionsweise von WEP

WEP basiert auf einem Geheimnis, das alle kommunizierenden Parteien (Clients und Access Points) teilen, dem WEP Schlüssel. Genaugenommen sind bis zu vier WEP

Schlüssel zu hinterlegen. Damit beide Parteien den gleichen wählen, ist im 32bit langen, nicht verschlüsselten, IV Feld des WEP Frames ein 2bit Feld für die Nummer des verwendeten WEP Schlüssels vorgesehen (vgl. Abbildung 2.2). Mit diesem geheimen symmetrischen Schlüssel werden Pakete vor dem Versenden verschlüsselt. Um eine Modifizierung während der Übermittlung zu erkennen, wird eine 32bit Prüfsumme gebildet.

Aus der Sicht des Versenders sind folgende Schritte durchzuführen, wobei die Einhaltung der hier gewählten Reihenfolge nicht verpflichtend ist.

1. Berechne einen CRC-Prüfsumme (cyclic redundancy check) über die Daten, den sogenannten ICV (integrity check value).
2. Wähle einen der vier zur Verfügung stehenden WEP-Schlüssel aus.
3. Generiere einen 24bit Zufallswert, den Initialisierungsvektor (IV).
4. Konkateniere 24bit IV und 40bit WEP-Schlüssel und gib dies als Eingabe in RC4, um den Schlüsselstrom zu erzeugen. Dieser wird mit einer (bitweisen) XOR-Verknüpfung auf die Daten und ICV angewendet.

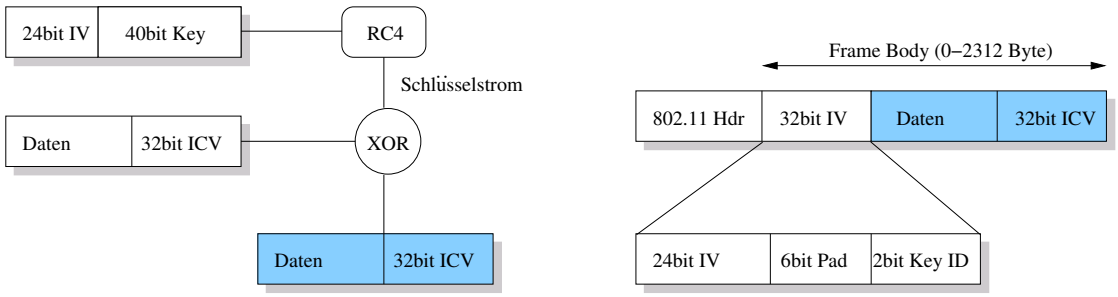


Abbildung 2.2: a) WEP Algorithmus b) WEP verschlüsselter Frame

Das 802.11 Paket, wie es über die Luftschnittstelle übertragen wird, hat dann den Aufbau gemäß Abbildung 2.2b. Der Empfänger extrahiert den IV, erzeugt dadurch den (gleichen) Schlüsselstrom mit RC4 (IV | Key) und wendet die XOR-Verknüpfung dieses Stroms auf (Daten | 32bit ICV) an. Abschließend wird die Prüfsumme CRC32(Daten) berechnet und mit dem übermittelten 32bit ICV verglichen. Bei Übereinstimmung wird das Paket akzeptiert.

2.5 Schwächen der Sicherheitsmechanismen in 802.11

In diesem Abschnitt werden die Mechanismen der von [802.11-99] abgedeckten Sicherheitsdienste jeweils beschrieben und direkt im Anschluss die Schwächen aufgezeigt.

2.5.1 Zugriffskontrolle

Der Mechanismus "Zugriffskontrolle" ist in 802.11 nicht vorgesehen. Wodurch Zugriffskontrolle auf jeden Fall nicht erreicht werden kann, sind folgende beiden Ansätze.

SSID,ESSID Um das eigene WLAN (BSS bzw. ESS) von anderen, im gleichen Bereich wirkenden WLANs unterscheiden und identifizieren zu können, kann ihm ein Name zugeteilt werden, wofür 32 Byte bereitgestellt sind. Ist diese Zeichenkette “Any”,

werden alle Verbindungen akzeptiert, ansonsten können nur Teilnehmer mit dem Access Point kommunizieren, die mit der gleichen SSID konfiguriert sind. So kann ein "zufälliges" Assoziieren vermieden werden.

Problem: Die SSID wird im Klartext übertragen. Sie befindet sich unter anderem in Beacon Frames, die vom Access Point ausgeschickt werden (vgl. Abbildung 2.3). Viele Geräte werden mit werkseitig aktiviertem Broadcast dieser Frames und einer herstellertypischen Default SSID ausgeliefert. Eine andere Möglichkeit, die SSID in Erfahrung zu bringen, ist das Senden eines Probe Requests, der von allen diesen Frame empfangenen Access Points mit einem Probe Response beantwortet wird.

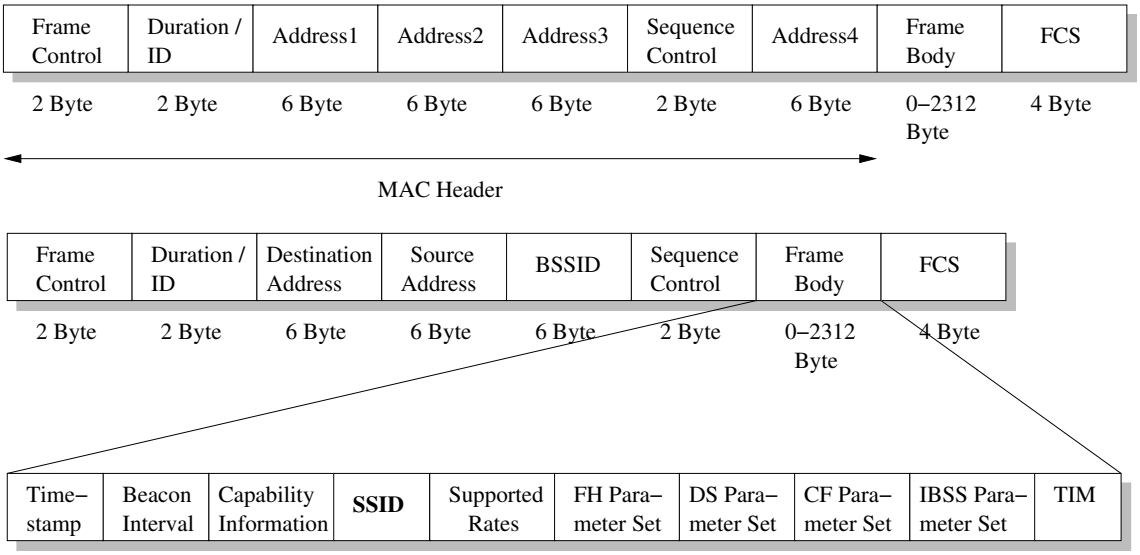


Abbildung 2.3: 802.11 Frame, Management Beacon Frame

MAC-Adressen Filter Die meisten Access Points bieten die Möglichkeit, Zugang über MAC Adressen zu beschränken. Dafür werden in der Regel zwei Listen mitgeführt, um explizit Geräteadressen zuzulassen oder explizit bestimmten Netzwerkgeräten den Zugang zu verbieten.

Problem: Dieser Schutz kann relativ einfach untergraben werden. Hierzu hat der Angreifer Pakete zugelassener Geräte abzufangen, die MAC Adresse herauszuziehen und diese dem eigenen Netzwerkadapter zuzuweisen. Dieses Verfahren wird auch MAC Spoofing genannt. Aus technischer Sicht ist dies keine Herausforderung.

2.5.2 Datenintegrität

Durch Berechnung des ICV, d.h. einer CRC32-Prüfsumme über die Daten soll deren Integrität geschützt werden. Der 32bit ICV wird den Daten vor der Verschlüsselung angehängt und mit diesen im Frame Body übertragen. Der Empfänger entschlüsselt den Payload, berechnet selbst die Prüfsumme CRC32 über die extrahierten Daten und vergleicht sein Ergebnis mit dem empfangenen ICV. Sollte also das Paket modifiziert worden sein, mutwillig oder aufgrund eines Übertragungsfehlers, so ist durch voneinander abweichende Prüfsummen eine Veränderung am Paket erkannt und es wird verworfen.

Problem: Die 32bit CRC Prüfsumme ist für diese Aufgabe ungeeignet. Sie ist ein fehlererkennender Code und wurde ursprünglich für die effiziente Erkennung von zufälligen Bitfehlern entwickelt, nicht aber, um als kryptographischer Hash oder Message Authentication Code (MAC) eingesetzt zu werden. Wie bei allen CRC Verfahren ist auch CRC32 mit der Eigenschaft behaftet, eine lineare Funktion zu sein, das heißt

$$\text{CRC32 (P1}\oplus\text{P2)} = \text{CRC32(P1)} \oplus \text{CRC32(P2)}$$

Das hat zur Folge, dass Bitunterschiede in der Checksumme mit Bitunterschieden in den (WEP-verschlüsselten) Daten in Beziehung zueinander stehen. Dem Angreifer ist es möglich, die Bits der CRC32 Prüfsumme zu bestimmen, die zu verändern sind, wenn beliebige Bits in den Daten verändert werden, so dass das resultierende Paket vom Empfänger als gültig akzeptiert wird.
Diese Schwäche wird erstmals in [Bo01] genannt und sollte auch bei Detailfragen bezüglich der Korrelation herangezogen werden, da auf diese sehr anschaulich und ausführlich eingangen wird. In den meisten Fällen gehen Angriffe auf diesen Sicherheitsmechanismus nicht über einen Denial-of-Service Charakter hinaus. Eine zu schützende Kommunikation ist in der Regel durch Mechanismen auf höheren Schichten gesichert, beispielsweise einer TLS-gesicherten HTTP-Verbindung. Wenn auch nicht auf Sicherungsschicht, so werden hier spätestens die gefälschten Pakete auch als solche erkannt und zurückgewiesen. Somit ist das Ausnutzen der CRC-Schwäche eher theoretischer Natur. Zu den Designzielen von 802.11i gehört es aber auch, bereits solche Fälle auszuschließen.

2.5.3 Authentifizierung

Für die Authentifizierung stehen in [802.11-99] zwei Varianten zur Verfügung, und zwar die Open System und Shared Key Authentifizierung. Erstere ist im Grunde ein Null-Algorithmus, da ein Client, der dieses Verfahren für seine Teilnahme am Netzwerk initiiert, ohne weitere Überprüfung den Status "authentifiziert" erhält (Abbildung 2.4). Es ist zu lesen, dass diese Variante in den Standard aufgenommen wurde, um die 802.11 Zustandsmaschine einfach zu halten. Auf den vom Client geschickten Authentication Request antwortet der Access Point mit dem Authentifizierungsergebnis "successful", wodurch eine gegenseitige Authentifizierung ([802.11-99], 8.1.1.2) erfolgt sein soll.

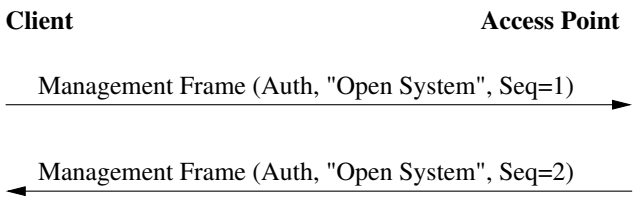


Abbildung 2.4: Open System Authentifizierung

Shared Key Authentifizierung ist nur bei Verwendung von WEP möglich, da es auf dem WEP Schlüssel basiert. Es ist ein einfaches Challenge Response Verfahren, in welchem der Client dem Access Point beweist, in Kenntnis des WEP Schlüssels zu sein. Zu diesem Zweck generiert der Access Point einen 128 Byte Challenge Text und sendet diesen an den Client, welcher daraufhin einen beliebigen 24bit IV auswählt, den Challenge verschlüsselt und beides als Challenge Response zurückschickt. Der Access Point entschlüsselt die Nachricht und extrahiert den Challenge. Stimmt dieser mit dem in Nachricht 2 gesendeten

überein, hat sich der Client erfolgreich authentifizieren können und darf an der Kommunikation teilnehmen.

Problem: Ein Angreifer kann durch Abhören und Herausfiltern des Challenge (Klartext), der Challenge Response (Ciphertext) und des IV eine erfolgreiche Authentifizierung herbeiführen, ohne in Kenntnis des WEP Schlüssels zu sein. Ein Angreifer besitzt durch Abfangen der Pakete dieser Konversation alle erforderlichen Informationen, um den RC4 Schlüsselstrom daraus ableiten zu können: Die Challenge Response ist doch Challenge $\oplus$ RC4(IV,Key), so dass Challenge Response $\oplus$ Challenge = RC4(IV,Key) ist. Von nun an kann er also als legitimierter Benutzer am WLAN teilnehmen. Dadurch, dass der Access Point nicht selbst den IV vorgibt sondern die Wahl dem Client überlässt, wird der abgefangene IV bzw. der Schlüsselstrom RC4(IV,Key) vom Angreifer einfach wiederverwendet. Der Angreifer wird erfolgreich authentifiziert, obwohl er das Geheimnis gar nicht kennt. Somit ist die Shared Key Authentifizierung völlig unsicher.

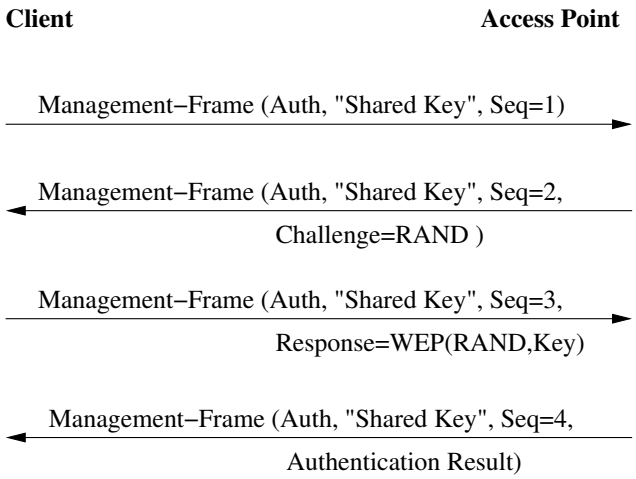


Abbildung 2.5: Shared Key Authentifizierung

Die eigentlich offentsichtliche aus der Shared Key Authentifizierung resultierende Unsicherheit ist aber lange Zeit nicht als solche identifiziert worden. Beispielsweise befindet sich im Intel Technology Journal des 2. Quartals 2000 ein Artikel ([Jtl00]), in welchem sich der Autor sehr zufrieden mit dieser Authentifizierung zeigt: "This [Anm.: Open Authentication] is usually implemented where ease-of-use is the main issue, and the network administrator does not want to deal with security at all. The shared key authentication approach provides a better degree of authentication than the open system approach." Wenige Monate später setzte dann die Aufdeckung der Schwächen von 802.11 ein.

Darüberhinaus weist die Authentifizierungsphase folgende, schwerwiegende Designprobleme auf.

1. Es findet keine Authentifizierung in der Richtung Benutzer zu Netzwerk (Access Point) statt, so dass für den Client keine Sicherheit besteht, auch mit dem richtigen Access Point zu assoziieren. Das Einbringen fremder Access Points, die im englischen Sprachraum gerne mit rogue (dt. "Schurken") APs bezeichnet werden, im Sinne eines Man-in-the-Middle Angriffs ist also möglich.

2. Aufgrund des Umstands, dass alle Stationen den gleichen Schlüssel verwenden, findet keine Identifizierung des Benutzers statt.

3. Die weithin akzeptierte Forderung von Schlüsselseparation wird nicht eingehalten, da für die Authentifizierung das gleiche Schlüsselmaterial wie für die Verschlüsselung herangezogen wird.

Aufgrund des eklatanten Fehldesigns der Shared Key Authentifizierung sind die Warnhinweise der WLAN Equipmentanbieter in Handbüchern oder Whitepapers, durch Wahl von Open System Authentication eine bessere Sicherheit zu erzielen, also durchaus berechtigt.

2.5.4 Vertraulichkeit

Wie in Abbildung 2.2 ersichtlich, soll die Vertraulichkeit der Daten durch den symmetrischen Stromchiffre RC4 erzielt werden. Die Daten und der ICV werden durch einen Schlüsselstrom, aus dem RC4 Pseudo-Zufallszahlengenerator stammend, mit XOR verschlüsselt. Als Eingabe dient der 40 bzw. 104bit[1] WEP Schlüssel und der 24bit Initialisierungsvektor IV. Der Empfänger muss im Besitz des WEP Schlüssels sein, um den Frame zu entschlüsseln. Unter der Annahme, dass jedes Paket mit einem anderen Schlüsselstrom verschlüsselt wird, ist dieses Verfahren als solches nicht schlecht. Weiterhin ist in der Wahl des Stromchiffres RC4 kein Fehler auszumachen. Die Sicherheitsdefizite ergeben sich nur aus der Art und Weise dessen Verwendung, und zwar durch die ungeschickte Konkatenation des WEP Schlüssels mit dem 24bit Zufallswert IV.

Problem 1: Zu kleiner Initialisierungsvektor

Die Serie von WEP Schwächen wurde im Herbst 2000 durch J.Walker eingeleitet, der die zu gering gewählte Länge des Initialisierungsvektors mit 24bit kritisierte. Nach hinreichend langem, passiven Aufzeichnen ist es möglich, den WEP Schlüssel zu ermitteln und somit jegliche in diesem WLAN stattfindende Kommunikation zu entschlüsseln.

Die Sicherheit von WEP basiert auf der Annahme, dass sich durch Änderung des IV der Schlüsselstrom RC4(IV,Key) ändert, mit dem der Payload des Pakets verschlüsselt wird. Somit sind selbst bei gleichem Klartext unterschiedliche Chiffretexte das Ergebnis. In der Regel wird der IV in jeder Runde, also pro Paket, um den Wert 1 inkrementiert. Somit dient die Hinzunahme eines Initialisierungsvektors der Erhöhung der Entropie des WEP-Schlüssels, wobei eine solche Zuführung mitunter auch als "Salz" bezeichnet wird.

Da der Adressraum für den IV mit 24bit gewählt ist, wiederholt sich der IV nach spätestens 2^{24}=16.7 Mio. Runden, was bei einem mäßig beanspruchten Netzwerk nach einigen wenigen Stunden der Fall ist, in der Praxis aber bereits wesentlich früher eintritt. Unter der realistischen Annahme, dass der WEP-Schlüssel nicht gewechselt wurde, gestaltet sich die Situation für zwei Klartextnachrichten P1 und P2

$$C1 \oplus C2 = P1 \oplus RC4(IV,Key) \oplus P2 \oplus RC4(IV,Key) = P1 \oplus P2$$

Der Schlüsselstrom wird aufgrund der Eigenschaft von XOR eliminiert. Ist einer der Klartexte, z.B. P1, bekannt, dann kann der andere durch

[1] Die Bitlänge der WEP Schlüssel wird oftmals missverstanden und in vielen Beschreibungen werden andere Bitlängen genannt. Der WEP Schlüssel hat eine Länge von 40bit, in der erweiterten, 1998 von Lucent eingeführten Fassung 104bit (vgl. [Wo03]). Die Verschlüsselung erfolgt hingegen durch den aus der Konkatenation WEP-Schlüssel | IV resultierenden Schlüsselstrom, was die Angabe 64 bzw 128bit rechtfertigt. Da der IV allerdings im Klartext übertragen wird, sollte man sich auf die Angabe der effektiven Schlüsselstärke (40/104) einigen.

$$(C1 \oplus C2) \oplus P1 = P2 \oplus P1 \oplus P1 = P2$$

aufgedeckt werden. Es kann weiterhin der zu dem Initialisierungsvektor gehörende Schlüsselstrom RC4(IV,Key) ermittelt werden. Da der Empfänger den Initialisierungsvektor für das Paket wissen muss, ist dieser unverschlüsselt mitzusenden und dadurch für einen Angreifer mühelos in Erfahrung zu bringen. Ein Angreifer würde also eine Tabelle anlegen und für jeden IV den zugehörenden Schlüsselstrom RC4(IV,Key) abspeichern, was ein Datenvolumen von etwa 24 GB erfordert. Durch Kenntnis des IV und des Schlüsselstroms ist der Angreifer in der Lage, beliebig Pakete einzuspielen, die vom Empfänger als korrekt erkannt werden. Quell- und Zieladresse werden in 802.11 Frames unverschlüsselt und kryptographisch ungeschützt übertragen, man kann also Nachrichten im Namen anderer schicken oder auch Nachrichten umlenken.

In dem bereits im vorigen Abschnitt referenzierten Artikel [Jtl00] wird bereits auf die Folge eines doppelt vorkommenden (IV,Key)-Paares hingewiesen, "If the same pair (Key, IV) is used for successive messages, this effect may reduce the degree of privacy. Changing the IV after each message is an easy way to preserve the effectiveness of WEP." Der Gedankenansatz geht erneut nicht weit genug, um an einen Überlauf des Zählers zu denken.

Problem 2: Schwache RC4 Schlüssel

Von Fluhrer et al. ist Mitte 2001 eine Schwachstelle im Design des Stromchiffre RC4 entdeckt worden ([Fl01]). Das Ausnutzen dieser Schwäche ist Ziel eines FMS-Angriffs, der in ungefähr 30 Minuten - durch aktive Erhöhung des Datenverkehrsaufkommens beschleunigt - den WEP-Schlüssel aufdecken kann. Unter bestimmten Bedingungen lassen sich nämlich Rückschlüsse auf den Schlüssel ziehen, mit dem der Schlüsselstrom generiert wurde, von dem ja bereits 24bit durch den Initialisierungsvektor IV bekannt sind. Beim FMS Angriff gilt das Interesse den von einer speziellen Klasse von IVs (sog. "weak IVs") verschlüsselten Paketen. Hinreichend viele derartige Pakete gesammelt lässt sich der WEP-Schlüssel nach und nach zurückgewinnen.

Im August 2001 setzten Stubblefield et al. ([St01]) diesen theoretisch beschriebenen Angriff erfolgreich in die Praxis um, worauf die meisten heute verfügbaren Cracktools wie Airsnort oder WEPcrack aufbauen. Da das Verfahren Byte für Byte des Schlüssels aufdeckt, wirkt sich eine Erhöhung der Schlüsselstärke nicht auf die Sicherheit aus, da es lediglich einen linearen Auffaktor der benötigten Zeit bedeuten würde. Wirkungsvoller ist, dass Hersteller ihre Firmware auf die FMS-Attacke anpassen, indem die aus den schwachen IVs resultierenden schwachen Schlüssel verworfen werden.

2.5.5 Administration

Der [802.11-99] Standard hat in keinerlei Hinsicht Aspekte dynamischen Schlüsselmanagements berücksichtigt, wodurch die Erzeugung, Verteilung und Erneuerung von Schlüsseln in den Aufgabenbereich des WLAN Betreibers fällt. Das statische Schlüsselmanagement ist äußerst nachteilhaft und viele Angriffe können durch diesen eklatanten Mangel erst realisiert werden. Je größer die Anzahl der beteiligten Geräte ist, desto komplizierter gestaltet sich die Verteilung eines neuen WEP Schlüssels, so dass es nicht ungewöhnlich ist, wenn größere Unternehmen den WEP Schlüssel selten bis gar nicht wechseln.

Auch gesetzt dem Falle, dass eine Kompromittierung erkannt (z.B. durch Verlust oder Diebstahl eines Geräts) und die Entscheidung getroffen worden ist, den WEP Schlüssel

auszutauschen, bleibt kaum etwas anderes übrig, als das Netz abzuschalten und den neuen Schlüssel - auf welchen Wegen auch immer - allen Geräten zukommen zu lassen. Spöttisch nennt [Ni02] als einzig praktikablen Weg der Schlüsselverteilung in großen, dynamischen 802.11 WLAN Umgebungen nur deren allgemeine Veröffentlichung.

2.6 Reaktion und Konsequenzen

> "The lesson is that security protocol design is very difficult, and is best performed with an abundance of caution, supported by experienced cryptographers and security protocol designers."
>
> Resumé zu WEP in [Cw03]

Im Rückblick ist es lohnenswert, nach den Ursachen und Gründen für die Verabschiedung eines derart mit Schwächen behafteten Standards zu suchen, wie es bei IEEE 802.11 der Fall ist. Insbesondere sollten bei der Entwicklung neuer Standards derartige Fehler nicht mehr unterlaufen. Die Schuld allein dem Kreis der Entwickler anzulasten, ist mit Sicherheit nicht sinnvoll, da eine Vielzahl äußerer Umstände das Entstehen von 802.11 in seiner 1999 verabschiedeten Fassung mit bestimmt haben. Als hilfreich erweist sich die Zusammenstellung der Eigenschaften des WEP Algorithmus, wie sie in [802.11-99], 8.2.2 erfolgt. Aufschlussreich ist weiterhin die im Februar 2001 erschienene Stellungnahme [Ke01] des 802.11 Working Group Chairman Stuart Kerry zu den bis dahin aufgedeckten Schwächen. Insgesamt ist das Ergebnis ein Zusammenwirken folgender Umstände:

1. An der Entwicklung des 802.11 Standards waren zumeist "volunteer engineers", also auf freiwilliger Basis arbeitende Ingenieure, beschäftigt. Der Großteil der Spezifikation beschäftigt sich mit den komplizierten technischen Aspekten der Übertragung von Datenpaketen über die Luftschnittstelle, nur elf der insgesamt 529 Seiten behandeln das Thema "Authentifizierung und Privatsphäre (privacy)".

2. Die IEEE verlangt für den vorzeitigen Zugriff auf einen Standard ein finanzielle Aufwendung in der Größenordnung um 80 US-$, was eine breite Beschäftigung während der Entwicklungsphase, insbesondere auch durch Kryptoanalysten, zumindest nicht begünstigte. Erst nach Ablauf von sechs Monaten nach Verabschiedung ist ein IEEE Standard frei verfügbar.

3. Im Zeitraum um 1999 war keineswegs absehbar, dass sich WLAN im Bereich der drahtlosen Netzwerke durchsetzen würde. Mehrere Konkurrenztechnologien wie HomeRF oder HiperLAN konkurrierten mit WLAN um die Vorherrschaft am Markt. Die relativ geringe anfängliche Datendurchsatzrate von 1-2 MBit/s erschwerte ebenfalls ein Aufkommen vermehrten Interesses. Die im Jahre 1999 verabschiedete Erweiterung erweiterte diese auf 11MBit/s, wobei die Sicherheitsmechanismen unverändert blieben.

4. Die Designkriterien von WEP weisen zudem darauf hin, dass WEP eine Sicherheit in vernünftiger Weise ("reasonably strong") biete, es die für ein auf Sicherungsschicht kritische Bedingung, selbst-synchronisierend zu sein, erfülle und durch die Wahl seiner Algorithmen, RC4 und CRC32, sehr effizient in Hardware wie Software zu implementieren sei.

5. Eine Verbreitung über die US-Grenzen hinaus kann nur mit Algorithmen geschehen, die vom US Department of Commerce für den Export freigegeben sind. Beispielsweise war 1999 nur der Export von 56bit Schlüsselstärke erlaubt.

In [Bo01] wird der Vorwurf laut, dass die meisten Fehler hätten vermieden werden können, da es fast ausschließlich Standardfehler gewesen seien. Ron Rivest, der Erfinder von RC4, beschreibt aus technischer Sicht das Fehlschlagen des WEP Algorithmus ([Rv01]):

> In protocols such as WEP, it is often necessary to generate different RC4 keys from different messages (or packets) from a common base key. A method frequently suggested to obtain the keys is to add or concatenate a counter to the base key. The key-scheduling algorithm of RC4 has been widely recognized to be rather lightweight for this purpose, particularly when the initial few bytes of plaintext are easily predictable. *RSA Security has discouraged such key derivation methods*, recommending instead that users consider strengthening the key scheduling algorithm by pre-processing the base key and any counter or initialization vector by passing them through a hash function such as MD5. Alternatively, weaknesses in the key scheduling algorithm can be prevented by discarding the first 256 output bytes of the pseudo-random generator before beginning encryption. Either or both of these techniques suffice to defeat the new attacks on WEP and WEP2.

Vergleicht man 802.11 WEP und IPsec, wie es damals bereits vorgelegen hat (Tabelle 2.1), lässt dies die Frage zu, warum in keinster Weise eine Orientierung an den von IPsec sorgsam ausgesuchten Techniken und Protokollen erfolgt ist.

Sicherheitsanforderung	IPSec	802.11
Wiedereinspielungsschutz	IPSec Sequenznummern	-
Geheimhaltung von Daten	DES, 3DES, IDEA, ...	RC4
Datenintegrität	MD5, SHA, Ripemd, ...	CRC32
Benutzerauthentifizierung	Passwort oder Zertifikat	-
Gegenseitige Authentifizierung	Preshared Keys, Public/Private Keys, Zertifikate	-
Schlüsselmanagement	PKI oder preshared Zertifikate / Master Keys	-
Dynamisches Rekeying	SA Lifetime; bei Überlauf von Sequenznummern	-
Externe Benutzerdatenbank	RADIUS, LDAP	-
Accounting, Monitoring	RADIUS, SNAP	-

Tabelle 2.1: IPsec vs. WLAN 802.11

2.7 Der neue Standard 802.11i

Im März 2001 wurde aus der seit 1999 existierenden Arbeitsgruppe "e", Quality of Service und Sicherheit, die Arbeitsgruppe "i" herausgelöst und mit der Überarbeitung der Sicherheitsmechanismen des 802.11 Standards beauftragt. Verglichen mit anderen ist es eine sehr aktive Arbeitsgruppe, indem in drei Jahren neun Draft Standards publiziert worden sind, was die Dringlichkeit der nötigen Verbesserungen deutlich macht. Bis 802.11i weitläufig in Implementierungen verfügbar ist und diese hinreichend interoperabel sind, wird allerdings noch einige Zeit vergehen.

Das neue Sicherheitssystem trägt den Namen RSN, Robust Security Network, und führt

- verbesserte Authentifizierungsmechanismen für Client und Access Point durch IEEE 802.1x,
- dynamisches Schlüsselmanagement und -Einrichtung durch IEEE 802.1x sowie

- stärkere Verschlüsselung durch TKIP und CCMP(AES)

ein, wobei diese Arbeit sich mit der Vorstellung der neuen Authentifizierungsmöglichkeiten beschäftigt. Der damit geschaffene gesamtheitliche Sicherheitskontext zwischen zwei Teilnehmern wird RSNA (RSN Association) genannt.

Es war von Vornherein bekannt, dass die Überarbeitung der Sicherheitsmechanismen bis zur Verabschiedung des 802.11i Standards mehrere Jahre in Anspruch nehmen würde. Um die Zeit bis dahin zu überbrücken, hat die Wi-Fi Alliance[2] im Jahre 2002 eine nicht mehr von Veränderungen bedrohte, sozusagen "stabile" Untermenge von 802.11i unter dem Namen WPA, (Wi-Fi Protected Access) auf den Markt gebracht. WPA wurde von Anfang an nur als Übergangslösung bezeichnet und war insbesondere ein Entgegenkommen an die Industrie, die nach den zahlreichen Katastrophenmeldungen aus den Jahren 2000 und 2001 den Einsatz von WLAN stark reduzierte.

Besondere Eigenschaften von WPA sind die Weiterverwendung der sich im Einsatz befindenen Hardware, da die verbesserten Sicherheitsmechanismen ausschließlich ein Software- oder Firmwareupgrade benötigen. WPA beinhaltet TKIP, 802.1x Authentifizierung sowie die Schlüsselhierarchie und das Schlüsselmanagement von 802.11i. Weiterhin bietet es für Endbenutzer und kleine Unternehmen eine Preshared-Key Lösung an, da in diesem Umfeld zumeist eine RADIUS Infrastruktur, wie sie für die Benutzerauthentifizierung gefordert ist, nicht existiert. Produkte, die seit September 2003 ausgeliefert werden, müssen WPA-fähig sein. Diese Überprüfung der jeweiligen Implementierung auf Interoperabilität führt die Wi-Fi Alliance durch und bescheinigt diese durch ein entsprechendes Zertifikat.

TKIP

Als Übergangslösung für verbesserte Verschlüsselung und Integritätssicherung ist das Temporal Key Integrity Protocol (TKIP) eingeführt worden. TKIP sollte gegen alle von WEP bekannten Angriffe resistent sein, dabei insbesondere die durch unsachgemäße Verwendung des RC4 Stromchiffres eingeführten Schwächen beseitigen. Gleichzeitig musste es die Bedingung erfüllen, mit bestehender, sich im Einsatz befindender Hardware kompatibel zu sein. Somit die Stromchiffre RC4 weiter im Einsatz geblieben. In der Tat findet sich in TKIP die WEP-Engine wieder (vgl. Abbildung 2.6), wobei folgende Erweiterungen gemacht wurden:

Message Integrity Code (MIC) Zur Erkennung von Fälschungen wird eine kryptographische Prüfsumme herangezogen (Michael). Sie hat als Eingaben die Quell- und Zieladresse sowie die Daten im Klartext. Als zusätzlicher Integritätsschutz dient zudem der 32bit WEP ICV. Der Empfänger berechnet selbst den MIC und prüft diesen auf Übereinstimmung mit dem empfangenen Wert. Je nach Ausgang wird das Paket als unverfälscht angenommen oder verworfen. Im letzten Fall leitet TKIP ein Rekeying ein. Dieses automatische Einleiten von Gegenmaßnahmen stellt ebenfalls eine Neuerung dar. Im Gegensatz zu bekannten MICs wie HMAC-SHA1 oder DES-CBC-MAC ist Michael äußerst performant und bietet gleichzeitig genügend Sicherheit, denn die Wahrscheinlichkeit, dass ein gefälschtes Paket nicht erkannt wird, ist bei 20bit Sicherheit, die Michael bereitstellt, 2^{-19}([Cw03]).

Packet Sequence Counter (PSC) Zur Aufdeckung von Wiedereinspielungen dient ein 48bit IV und ein IV Sequence Counter. Wenn beim Empfang eines fragmentierten

[2]Die Wi-Fi Alliance ist eine internationale Vereinigung von mehr als 200 Firmen aus dem WLAN Hardware und Software Bereich. Sie wurde 1999 gegründet, um 802.11 Hardware bezüglich ihrer Interoperabilität zu zertifizieren. Bis zur Umbenennung im Oktober 2002 trug sie den Namen WECA (Wireless Ethernet Compatibility Alliance).

Pakets der PSC außer der Reihe ist, wird es vom Empfänger verworfen. Die Erweiterung des IV-Raums auf 48bit soll sicherstellen, dass in einer Sitzung kein IV wiederverwendet wird und somit die Möglichkeit des Auftretens gleicher (IV,Key)-Paare ausgeschlossen wird.

Per-Packet Key Mixing Eine aus zwei Phasen bestehende Mixfunktion soll die Korrelation zwischen Initialisierungsvektor und schwachen Schlüsseln beseitigen. Damit wird dem FMS Angriff entgegnet, der diese Schwäche bei 802.11 WEP ausnutzt.

Rekeying Ist ein automatischer Mechanismus, um dynamisch frisches Schlüsselmaterial für Verschlüsselung und Integritätsschutz zu verteilen und tritt somit dem statischen Schlüsselmanagement von 802.11 entgegen. Da konstruktionsbedingt die Mixfunktion höchstens 2^{16} verschiedene IVs verarbeiten kann, ist demzufolge mindestens alle 65536 Pakete ein Schlüsselupdate einzuleiten.

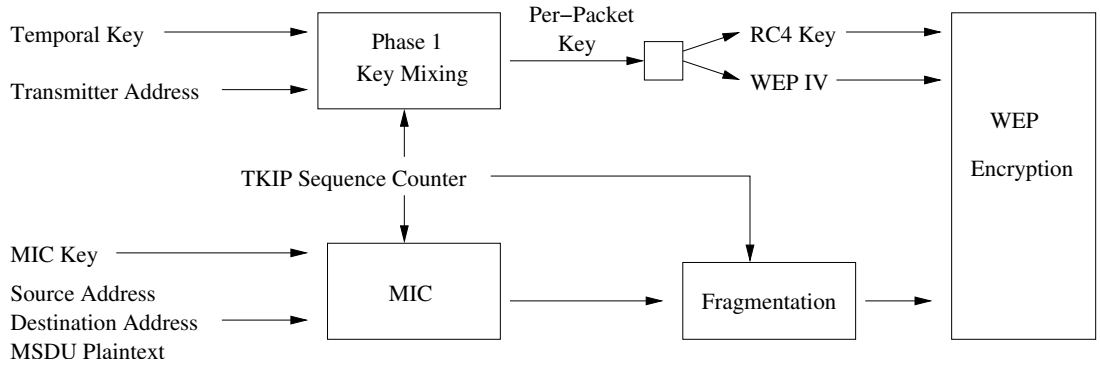


Abbildung 2.6: TKIP Verschlüsselungsprozess (nach [Ea03])

CCMP

Diese Bezeichnung steht für den von 802.11i empfohlenen und für ein RSN verpflichtenden Verschlüsselungsalgorithmus und ist die Abkürzung für "Counter Mode with Cipher Block Chaining - Message Authentication and Control Protocol". CCMP basiert auf dem CCM Modus des AES Algorithmus, wobei der Counter Mode der Vertraulichkeit und der CBC-MAC der Authentizität und Datenintegrität dienen soll.
Im Oktober 2002 wurde durch das National Institute of Standard and Technology (NIST) der Rijndael Algorithmus als Advanced Encryption Standard (AES) ausgewählt. Die Vorgaben an den Algorithmus waren, zukunftsweisend, gegenüber allen kryptographischen Angriffen resistent, vollständig analysiert und frei verfügbar zu sein. Bedingungen an die Schlüsselstärke und eine praktische Implementierbarkeit wurden ebenfalls gestellt. So verwendet CCMP einen 128bit Schlüssel mit einem 48bit Initialisierungsvektor für Wiedereinspielungsschutz.
In einem Zwischenstadium von 802.11i existierte zudem das WRAP (Wireless Robust Authenticated Protocol), das auf dem AES Offset Codebook Modus aufsetzt. Aufgrund Streitigkeiten im Urheberrecht und drohender Patentschutzverletzung wurde es aus dem Standard entfernt und durch CCMP ersetzt.
Für weitergehende Informationen zu TKIP und CCMP ist die Artikelserie [Wa02] oder auch die Ende 2003 an der TU Darmstadt erschienene Diplomarbeit [Ha03] zu empfehlen. Als Ausgangspunkt für Recherchen zu Rijndael eignen sich [Ni01] und [Ri01]. Fortschritte über kryptographische Analysen von AES sind auf der vom Kryptoanalysten Courtois betreuten Webseite [Co04] zu finden.

Kapitel 3

Authentifizierung in WLAN 802.11i

Für die Bereiche Authentifizierung, Zugangs- bzw. Zugriffskontrolle und Schlüsselverteilung wird in 802.11i die IEEE 802.1x Architektur herangezogen, wobei sich der im Jahre 2001 verabschiedete Standard [802.1x] zur Zeit in einer Überarbeitungsphase befindet. Ursprünglich für den Einsatz in Ethernet, Token Ring und FDDI ausgelegt sind Erweiterungen nötig, um in der gesamten IEEE 802 LAN Architektur, insbesondere in 802.11 Wireless LAN, angewendet werden zu können. Kurz umrissen wird mit 802.1x ein Protokoll definiert, um nichtauthorisierte Geräte wie Benutzer an einem Zugriff über einen öffentlich zugänglichen Port auf ein lokales Netz zu hindern.

Zunächst wird das 802.1x Rollenmodell mitsamt seiner Terminologie eingeführt. Darin werden drei Parteien definiert und ihre Aufgaben hinsichtlich der Authentifizierung und der Zugriffskontrolle aufgezeigt. Im zweiten Schritt wird auf die Nachrichten, die in diesem Zusammenhang zwischen den Parteien ausgetauscht werden, eingegangen. Da 802.1x lediglich ein Rahmenwerk darstellt, ist ein Protokoll zum Transportieren der Daten zwischen den verschiedenen Netzwerkkomponenten notwendig, wofür 802.1x das Extensible Authentication Protocol (EAP) gewählt hat. Es selbst ist wiederum nur ein Rahmenwerk und lässt verschiedene konkrete Authentifizierungsverfahren zu, die zum Beispiel auf Benutzername/Passwort, auf Zertifikaten oder auf Smartcards beruhen. Die Beschreibung einiger ausgewählter Methoden samt der wichtigsten Eigenschaften erfolgt in Kapitel 4.

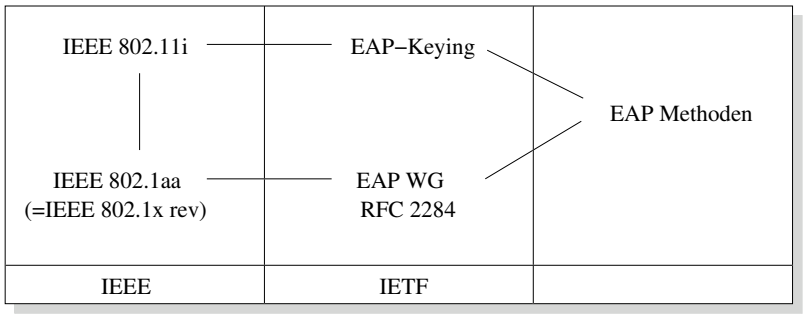


Abbildung 3.1: IEEE 802/IETF Liaison

Die Abhängigkeiten, die sich durch diese Konstellation zwischen IEEE und IETF ergeben, veranschaulicht Abbildung 3.1. Um die Authentifizierungsphase in den Gesamtablauf einordnen zu können, bietet sich an dieser Stelle ein Überblick über die vier Kommunikationsphasen in 802.11i an (Abbildung 3.2). Obwohl demnach nur Phase 2 Gegenstand

dieser Arbeit ist, werden Zusammenhänge zu den anderen Phasen soweit es nötig oder dem Verständnis dienlich ist, aufgezeigt.

Phase 1	Entdeckung und Assoziierung
Phase 2	Authentifizierung
Phase 3	Schlüsselaustausch
Phase 4	Verschlüsselte Kommunikation

Abbildung 3.2: Kommunikationsphasen im WLAN

3.1 Portbasierte Kontrolle des Netzwerkzugangs

In dem typischen Einsatzbereich von 802.1x erfordert das Netzwerk vom Benutzer eine Authentifizierung, bevor irgendein anderer Datenaustausch möglich ist, z.B. bevor über DHCP dem Client eine IP Adresse zugeteilt wird. Die Zugangskontrolle erfolgt somit bereits am Rand des Netzwerks.

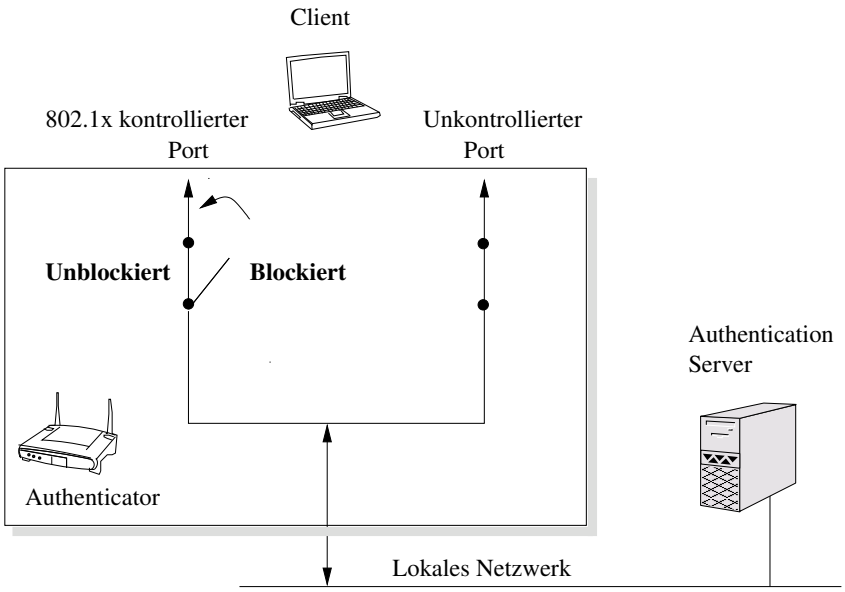


Abbildung 3.3: Rollenmodell in der 802.1x Architektur

Port Darunter ist ein allgemeiner Zugangspunkt zum Netzwerk zu verstehen. Bei 802.1x-fähigen Ethernet-Switches ist mit einem Port in der Tat ein Kabelsteckplatz gemeint, bei Wireless LANs hingegen die Assoziation zwischen Access Point und mobilem Endgerät. Der Zustand eines 802.1x Ports wird mit den Attributen kontrolliert, unkontrolliert, blockiert und unblockiert angegeben. Nur die Kombination (unkontrolliert, blockiert) ist nicht zulässig, da der unkontrollierte Port für einen Austausch von Authentifizierungsnachrichten immer geöffnet ist.

Supplicant Der Supplicant, auch Client genannt, ist das Gerät, das sich authentifizieren muss. Dafür übermittelt dieser dem Authenticator seinen Berechtigungsnachweis (credential), das zum Beispiel Benutzername und Passwort oder ein Zertifikat sein kann, und fordert hierdurch Zugang zum Netzwerk.

Authenticator Als Authenticator wird das Gerät bezeichnet, über dessen 802.1x-kontrollierte Ports dem Client der Zugang zum Netzwerk ermöglicht wird. Die eigentliche Entscheidung hierüber fällt aber nicht der Authenticator, sondern eine dritte Instanz, der Authentication Server. An diesen wird die Anfrage des Clients weitergeleitet. Der Nachrichtenaustausch während der Authentifizierungsphase findet zwischen Supplicant und Authentication Server statt. Der Authenticator agiert in dieser Zeit lediglich passiv als passthrough-device. Am Ende dieser Phase wird der 802.1x-kontrollierte Port freigegeben oder bleibt im blockierten Zustand, abhängig von der Entscheidung des Authentication Server über den Erfolg der Verhandlung.

Authentication Server Der Authentication Server bietet dem Authenticator einen Dienst zur Authentifizierung an. Anhand des vom Supplicant erhaltenen Identitätsbeweises entscheidet er darüber, ob der Supplicant auf die Dienste, die der Authenticator zur Verfügung stellt, zugreifen darf. Dem Authentication Server obliegt zudem die Auswahl der genauen Authentifizierungsmethode. Ist im weiteren Teil der Arbeit von Server, EAP-Server oder AAA-Server die Rede, so ist diese Einheit gemeint.

In Abbildung 3.3 werden die drei Parteien in Beziehung zueinander gebracht. Im Rahmen von Wireless LAN, das im Infrastruktur Modus betrieben wird, ist der Supplicant das mobile Gerät, beispielsweise ein Notebook, PDA oder drahtloses VoIP Telefon, der Authenticator der Access Point und der Authentication Server ein AAA Server (s.u.) im LAN. Gleichwohl ist die Trennung von Authenticator und Authentication Server in zwei Geräten nicht verpflichtend - sie können auch gemeinsam im Access Point untergebracht sein, auch wenn diese Variante in der Praxis selten vorkommt.

Zwischen Supplicant und Port sowie zwischen Port und Authenticator Zustandsmaschine liegt eine 1:1 Beziehung vor. Ein Authenticator hat eine (theoretisch) unbegrenzte Anzahl von managebaren Ports.

Alle den Nachrichtenaustausch der Authentifizierung betreffenden Nachrichten werden über den unkontrollierten Port geleitet. Dieser ist immer geöffnet. Der kontrollierte Port ist ausschließlich für das Passieren allgemeinen Datenverkehrs zwischen Supplicant und Authenticator vorgesehen und befindet sich solange im blockierten Zustand, bis die über den unkontrollierten Port erfolgte Authentifizierung erfolgreich abgeschlossen ist. Solange keine erfolgreiche Authentifizierung erfolgt ist, werden alle Nachrichten außer EAPOL und EAP verworfen.

3.2 RADIUS als Authentication Server

Authentifizierung von Netzwerkgeräten auf Sicherungsschicht ist bereits länger im Einsatz, und zwar mit Einwahlverbindungen, beispielsweise in das Internet oder in ein Firmennetzwerk. Das RADIUS (remote access dialin user service) Protokoll wurde in seiner ursprünglichen Form genau hierfür geschaffen, indem es Dienste bereitstellt für zentralisierte Authentifizierung, Authorisierung und Accounting (AAA).

Damit schafft es Abhilfe, nicht auf jedem Einwahlserver Benutzerlisten verwalten zu müssen. Der Client wählt mit einem Modem über die herkömmliche Telefonleitung einen Einwahlserver (Network Access Server; NAS) an, der die Benutzeridentifizierungsdaten zu einem zentralisierten RADIUS Server weiterleitet. Dieser hat Zugriff auf eine Benutzerdatenbank und genauere Angaben, welche Dienste welchem Benutzer zur Verfügung stehen.

Nach der Verifizierung der Zugangsdaten teilt dieser dem Einwahlserver in einer Antwort den Ausgang der Authentifizierung mit, ob dem Benutzer also der Zugang zum Netz gewährt oder verweigert werden soll. Wegen der Client-Server Anordnung zwischen NAS und RADIUS Server wird der NAS oft als RADIUS-Client bezeichnet.

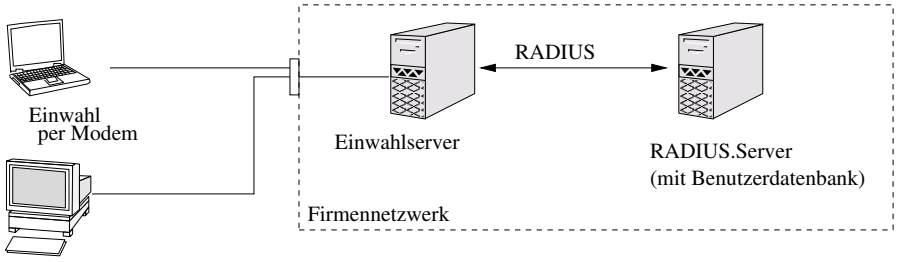


Abbildung 3.4: RADIUS in Firmennetzwerken

Von nun an wird die Wahl von RADIUS als AAA-Protokoll angenommen. Die IEEE 802 Architektur darf natürlich keine Vorgaben hinsichtlich höherer Protokolle wie TCP/IP machen, sondern kann nur Empfehlungen aussprechen. So legt [802.11i] auch RADIUS als die derzeit geeigneteste Wahl eines AAA Protokolls nahe. Aus diesem Grunde ist auch die Spezifikation der für den Einsatz in 802.11 erforderlichen RADIUS Erweiterungen am weitesten fortgeschritten. Als Alternative wird noch Diameter gehandelt, das als Next-Generation AAA-Protokoll zum einen die Schwächen von RADIUS (Beschränkung der Attributgröße auf maximal 255 Byte, keine Ende-zu-Ende Sicherheit usw.) und zum anderen die neuen Anforderungen durch Roaming und mobile IPv6 Netzwerke adressiert. Für weitere Informationen zu Diameter kann z.B. [Ve02] dienen.

3.3 Extensible Authentication Protocol (EAP)

3.3.1 Beschreibung

802.1x verwendet das Extensible Authentication Protocol (EAP), um Authentifizierungsnachrichten zwischen den verschiedenen Netzwerkkomponenten, also Supplicant, Authenticator und Authentication Server, zu verschicken. EAP ist in RFC 2284 als Standard definiert und war ursprünglich für den Einsatz in PPP konzipiert. Da in PPP jede Authentifizierungsmethode eine eigene Protokollnummer benötigt, sollte einem möglichen Überhandnehmen eine generische Methode hinzukommen, in die als Rahmenwerk weitere Methoden eingebettet werden können.

Da EAP in der Sicherungsschicht arbeitet, steht neben der Kapselung in PPP, für das es ursprünglich entwickelt wurde, ein Transport über 802.3 Ethernet oder 802.11 WLAN prinzipiell nichts im Wege. Bis Mitte Februar 2004, als [RFC2284-] von der IETF als Standard verabschiedet wurde, hat sich EAP in einer längeren Überarbeitungsphase befunden, um an die Bedürfnisse von WLAN angepasst zu werden.

Die wichtigsten Eigenschaften von EAP sind:

- Flexibilität: Da EAP lediglich ein Rahmenwerk für Authentifizierung im Netzwerkbereich beschreibt, sind neue Authentifizierungsmethoden lediglich als neuer EAP Typ hinzuzufügen, wofür die Reservierung einer Protokollnummer (siehe Anhang D) ausreicht. Der Authenticator erkennt ein eintreffendes Paket als EAP Paket und leitet es an

den Authentication Server bzw. an den Supplicant weiter. Somit hat eine Interpretation des Inhalts nicht zu erfolgen, was den Vorteil mit sich bringt, dass der Authenticator bei einer neuen EAP Methode kein Soft- bzw. Firmwareupgrade benötigt.

- Medienunabhängigkeit: EAP agiert auf Sicherungsschicht und ist somit nicht auf höhere Protokolle wie IP angewiesen. Der Transport von EAP über UDP (L2TP), IKEv2 und TCP (PIC) ist aber auch spezifiziert.
- Fehlerkorrektur: EAP gewährleistet den zuverlässigen Transport des Authentifizierungsprotokolls, es bietet Elimination von Duplikaten und unterstützt erneutes Übertragen (retransmissions). Hingegen bietet EAP keinen Mechanismus für Fragmentierung und Wiederzusammensetzen. Generiert eine EAP Methode Payloads größer als 1020 Byte, was der minimalen EAP MTU entspricht, muss sie selbst die Fragmentierung handhaben.

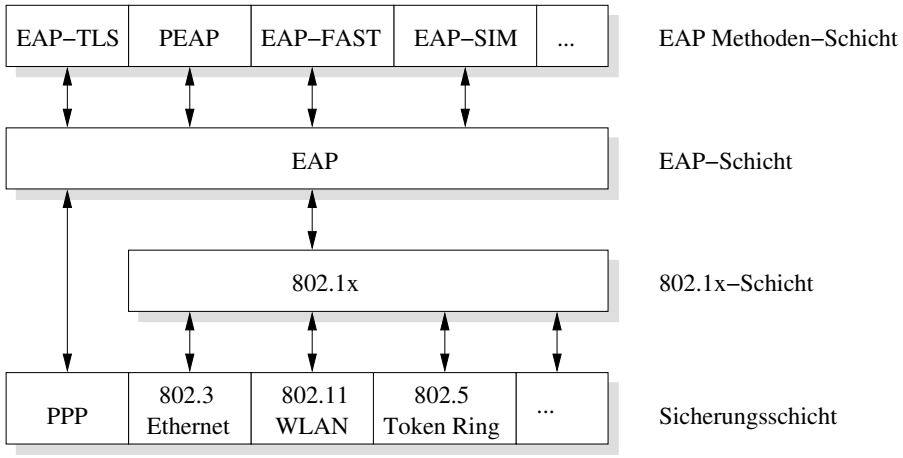


Abbildung 3.5: EAP in der 802.1x Architektur

Der 802.1x Standard definiert ein Protokoll, wie EAP Pakete über IEEE 802 zu verschicken sind. Zu diesem Zweck wird EAPOL (EAP over LAN) eingeführt, wobei ein EAPOL Paket genau ein EAP Paket kapseln kann. Authentifizierungsverfahren außerhalb von EAP, wie PAP oder CHAP, werden von 802.1x nicht unterstützt. Die Rolle, die RFC 2284 zukommt, ist also, Nachrichtenformate und -bedeutungen zu definieren sowie den Nachrichtenaustausch zwischen den drei Parteien zu spezifizieren.

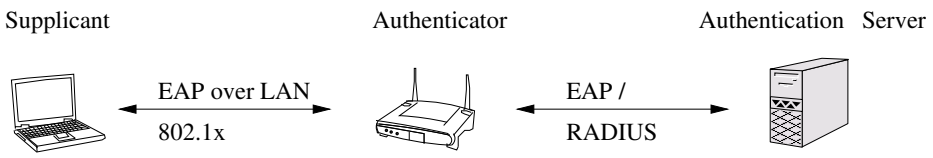
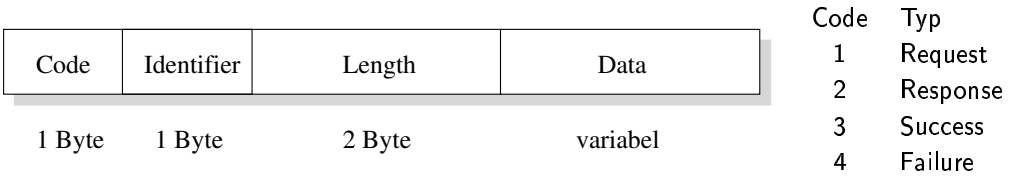


Abbildung 3.6: Transport von EAP

Im folgenden Abschnitt soll ein Überblick über die verschiedenen Nachrichten von EAP und deren Transport in 802.1x sowie RADIUS geschaffen werden. Ihre Kenntnis ist grundlegend für eine Beschreibung des EAP Nachrichtenaustauschs in der Authentifizierungsphase.

3.3.2 EAP Paketformate

Das EAP Protokoll ist ein Client-Server Protokoll, der Nachrichtenaustausch besteht aus Request und Response Paaren. Durch das Code Feld wird der Typ des EAP Pakets identifiziert, wobei vier Typen vorgesehen sind. Davon abhängig werden auch die Daten interpretiert. Der Identifier dient dazu, zusammengehörende Requests und Responses zu ermitteln. Das 2 Byte Length Feld repräsentiert die Gesamtgröße des EAP Pakets, ist also die Summe über Code, Identifier, Length und Data. Ist die darin angegebene Länge größer als die empfangenen Byte, so ist das Paket stillschweigend zu verwerfen, ist sie kleiner als die empfangenen Byte, so wird das Paket angenommen und die überschüssigen Byte als Padding der Sicherungsschicht angesehen und ignoriert.



Code	Typ
1	Request
2	Response
3	Success
4	Failure

Abbildung 3.7: EAP Paket

Request und Response Pakete (Code 1 und 2):
Requests werden ausschließlich in der Richtung Authenticator zu Client, Responses ausschließlich in der Richtung Client zu Authenticator verschickt. Um den Grund des Requests näher zu spezifizieren, sozusagen über die Forderung Aufschluss zu geben, ist zwischen Length und Data der Type in 1 Byte Länge eingeschoben. Jeder vom Authenticator verschickte Request wird mit einem eindeutigen Identifier versehen. Dies ist auch überaus wichtig, da ein Authenticator auf viele gleichzeitig stattfindende Authentifizierungen ausgelegt sein muss. Empfängt nun der Authenticator eine Response mit unbekanntem Identifier, d.h. dass von seiner Seite kein zugehöriger Request vorausgegangen ist, wird das Paket verworfen. Trifft eine Response nicht ein, so wird der Request wiederholt, und solange der Handschlag nicht abgearbeitet ist, darf kein anderer Request eingeleitet werden.

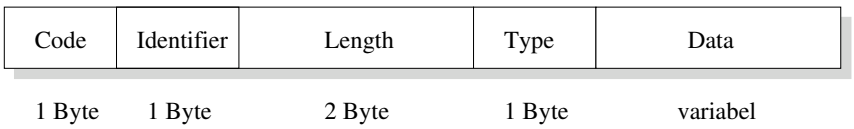


Abbildung 3.8: EAP Request, Response

Weitere Vorgaben und Empfehlungen zur Verwendung des Identifiers, nicht zuletzt aufgrund des kleinen Adressraums von einem Byte, sind in [RFC2284-] Abschnitt 3.4 zu finden. Das Type Feld bezeichnet die Art des Requests bzw. der Response. Die Werte 4-253 sind für EAP Methoden reserviert, wobei bis zum jetzigen Zeitpunkt bis einschließlich 44 allokiiert sind (s. Anhang D), wovon die IANA bereits 20 als wiederverwendbar ausgeschrieben hat ([17], 2.1). Sollte die Anzahl der EAP Methoden diesen Bereich ausfüllen, so werden sie unter 254, Expanded Types, weitergeführt. Für experimentelle oder noch nicht registrierte EAP Methoden steht der Code 255 zur Verfügung, den Implementierungen auch mit gewissen Freiheiten behandeln sollten.

Type		Type	
1	Identity	4-253	EAP-Methoden
2	Notification	254	Expanded Types
3	Nak (Response Only)	255	Experimental Use

Tabelle 3.1: EAP Request, Response Typen

Jede EAP Implementierung muss mindestens die ersten vier Typen bereitstellen, das heisst die drei speziellen Typen Identity, Notification und Nak sowie das CHAP-Äquivalent EAP-MD5.

Identity Diese Nachricht wird nur zu Beginn der Authentifizierung benötigt, womit der Client seine Identität zustellt. Da zu diesem Zeitpunkt noch keine Verschlüsselung erfolgt und die Nachricht von jedem ausgelesen werden kann, muss sie wohlgemerkt nicht die echte Identität beinhalten (vgl. Abschnitt 3.4).

Notification Zum Versand nebenläufiger Mitteilungen des Authenticators an den Client ist der Notification Typ heranzuziehen, auf welche der Client mit einer EAP-Response/Notification antworten muss. 1015 Bytes können an Nachrichtentext übertragen werden, da 1020 Bytes als minimale MTU in EAP vorgegeben sind. Aus bestimmten Gründen kann eine EAP Methode diesen Nachrichtentyp aber auch verbieten. Dieser Nachrichtentyp beinhaltet keine authentifizierungsrelevanten Informationen sondern hat rein informative Absichten, zum Beispiel den Hinweis auf ein bald ablaufendes Passwort oder die Zustellung von Warnmeldungen.

Nak Abkürzung für Not Acknowledged, kann wie Typ 254 nur in Response Paketen auftreten. Hiermit kann der Client den Authentication Server informieren, dass er dessen vorgeschlagene EAP Methode nicht unterstützt.

Success und Failure Pakete (Typ 3 und 4):
Ist die Authentifizierungsphase erfolgreich verlaufen, wird dem Client dieser Ausgang in einer EAP-Success Nachricht mitgeteilt. Wenn diese Phase nicht erfolgreich endet, was aus vielfältigen Gründen passieren kann, erscheint eine EAP-Failure Nachricht. Der Paketaufbau ist in diesem Falle einfacher, Data und Type Feld gibt es nicht und Length beinhaltet die Konstante "4" Byte.
Im Gegensatz zu EAP-Requests, die solange wiederholt werden, bis eine zugehörende EAP-Response eintrifft, werden EAP-Success und EAP-Failure nicht erneut gesendet - hier muss die Implementierung mit Timeouts operieren und der Situation auf geeignete Weise entgegenwirken.

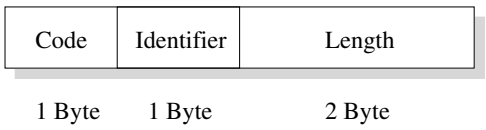


Abbildung 3.9: EAP Success, Failure

3.3.3 EAP über 802.1x

Dieser Abschnitt beschreibt, wie EAP Nachrichten in dem dafür vorgesehenen 802.1x EAPOL Paket gekapselt werden. In Abbildung 3.10 ist der Aufbau eines EAPOL Pakets

dargestellt, woraus ersichtlich wird, dass es selbst noch einen Träger zum Transport benötigt. In 802.11 WLAN geschieht dies durch Data Frames und wohlgemerkt nicht, wie bei 802.11 Open System bzw. Shared Key Authentifizierung der Fall, in Management Frames.

Als Ethernettyp für 802.1x ist 0x888E zugewiesen, die Protokollversion in [802.1x] die Konstante 1, in [802.1x-9] die Konstante 2. Es gibt fünf verschiedene EAPOL Pakettypen (s. Tabelle 3.2). Mit Code 0 wird festgelegt, dass im Packet Body genau ein EAP Paket gekapselt ist. Diese Pakete sind also während des Authentifizierungsvorgangs in der Unterhaltung zwischen Supplicant und Authenticator anzutreffen. Den Beginn und das Ende einer 802.1x-authentifizierten Sitzung bilden EAPOL-Start bzw. EAPOL-Logoff, Fehler werden durch EAPOL-Alert Nachrichten verschickt. Ihre Gemeinsamkeit besteht darin, keinen Packet Body mitführen. Da ASF-Alerts mehr Gefahren als Nutzen bringen, werden diese von RSN nicht verwendet. Es gibt keinen Grund für einen nicht authorisierten Supplicant, Fehlermeldungen irgendeiner Art an das System zu senden, somit sind ASF-Alerts von der Implementierung des Authenticators abzufangen und stillschweigend zu verwerfen.

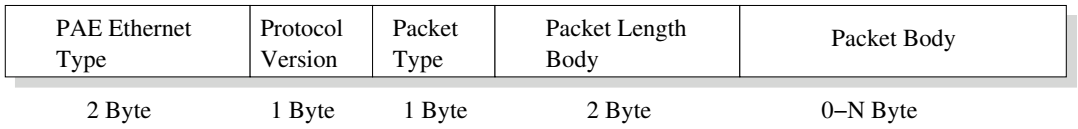


Abbildung 3.10: EAPOL Paket

Code	Typ
0	EAP Paket
1	EAPOL-Start
2	EAPOL-Logoff
3	EAPOL-Key
4	EAPOL-Encapsulated-ASF-Alert

Tabelle 3.2: EAPOL Pakettypen

Wenn der Supplicant verfügbare Authenticator, also Access Points, in seiner Umgebung in Erfahrung bringen will, wird dies durch Senden von EAPOL-Start erreicht. Diese Nachricht wird an eine spezielle Gruppen-Multicast MAC Adresse adressiert, 00:80:C2:00:00:03, die für Authenticator reserviert ist.

Das EAPOL-Key Paket wird erst nach abgeschlossener, erfolgreicher Authentifizierung benötigt, wenn zwischen Supplicant und Authenticator Schlüsselmaterial generiert und abgeglichen wird. Da der Authenticator während der Authentifizierung lediglich für das Durchleiten der Nachrichten verantwortlich war, muss diesem zuvor vom Authentication Server entsprechendes Schlüsselmaterial zugestellt werden.

Nach erfolgreicher Authentifizierung sind Client und Authenticator gleichermaßen im Besitz eines langlebigen Schlüssels. Aus diesem wird im nächsten Schritt kurzlebiges Schlüsselmaterial abgeleitet, das für die Verschlüsselung und Integritätssicherung vorgesehen ist. Natürlich wird zu keinem Zeitpunkt irgendein Schlüssel direkt über

die Luftschnittstelle übertragen, vielmehr nur zu deren Ableitung dienende Teilinformationen, z.B. Zufallszahlen (Nonces), ähnlich wie beim Diffie-Hellman Schlüsselaustausch.

Um aber auch den Transport der Nonces vertraulich und integritätgeschützt abzusichern, werden EAPOL-Key Pakete verwendet, deren Packet Body genau ein Key-Descriptor einschließt. Der 1 Byte Descriptor Type kann zwei Werte annehmen:

- Descriptor Type = 1: RC4 Key Descriptor. Er wird in der Überarbeitung [802.1x-9] nur noch aus Kompatibilitätsgründen mitgeführt, die ihn aber nicht mehr unterstützt (deprecated). In 802.11 wird der RC4 EAPOL-Key Frame zur Verteilung von Unicast ("key mapping") und Multicast ("default") Schlüsseln für WEP herangezogen (vgl. u.a. RFC 3580, Abschnitt 4).

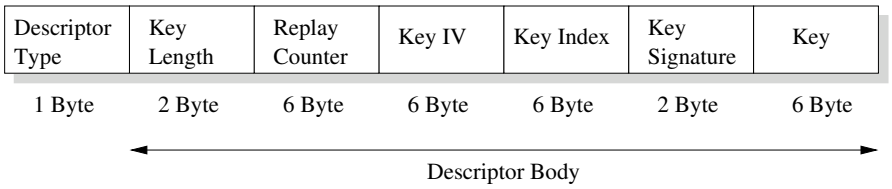


Abbildung 3.11: RC4 Key Descriptor

- Descriptor Type = 2: 802.11 Key Descriptor. Der in Abbildung 3.12 dargestellte Aufbau des Key Descriptors macht deutlich, dass im Schlüsselmanagement von 802.11i wesentliche Erweiterungen erfolgt sind, um den an sie gestellten Anforderungen gerecht zu werden.

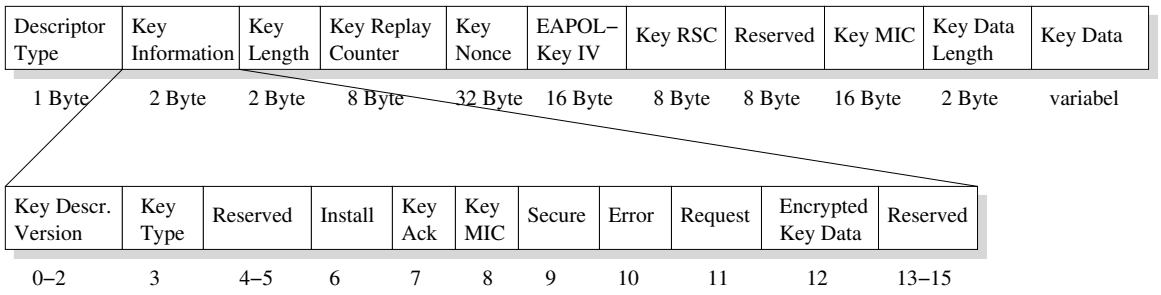


Abbildung 3.12: 802.11 Key Descriptor

Mit der Definition des Key-Descriptor Formats endet auch der Zuständigkeitsbereich von IEEE 802.1x. Die genauen Aufgaben werden in [802.11i] spezifiziert. Im RSN wird ausschließlich der Key-Descriptor vom Typ 2 verwendet, da der RC4 Key-Descriptor nur dynamisches WEP-(Re)Keying abdecken kann.

3.3.4 EAP über RADIUS

Wie EAP mit RADIUS transportiert wird, ist in der RADIUS Erweiterung RFC 2869 erläutert, die in der Zwischenzeit eine Überarbeitung erfahren hat und im September 2003 durch RFC 3579 abgelöst wurde. Nachrichten werden in RADIUS Attributen transportiert, so ist also eine RADIUS Erweiterung nichts anderes als das Hinzufügen

neuer Attribute.

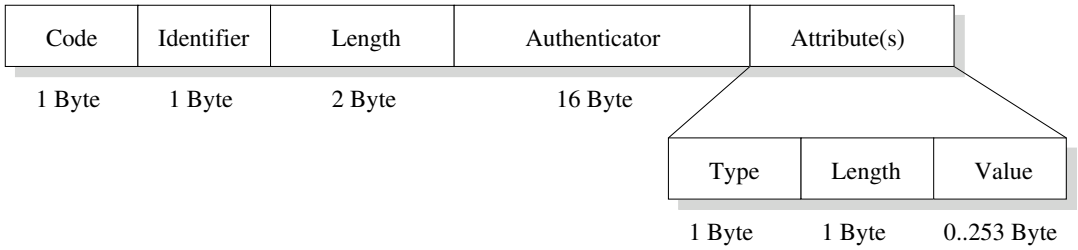


Abbildung 3.13: RADIUS Paket

Zunächst soll das allgemeine RADIUS Paket betrachtet werden. Sein Aufbau ist sehr einfach und erinnert an die Leichtgewichtigkeit von EAP Paketen. Über den Inhalt des 1 Byte langen Code Felds wird der Typ der RADIUS Nachricht definiert. Der Identifier dient dem Zusammenführen von Requests und Challenges und das Länge Feld beinhaltet die Gesamtgröße des Pakets in Byte.

Code	Typ	Richtung
1	Access-Request	Authenticator → RADIUS Server
2	Access-Accept	RADIUS Server → Authenticator
3	Access-Reject	RADIUS Server → Authenticator
11	Access-Challenge	RADIUS Server → Authenticator

Tabelle 3.3: RADIUS Nachrichten

In Tabelle 3.3 sind die vier für EAP in RADIUS relevanten Nachrichten aufgelistet, wobei Access-Request Nachrichten ausschließlich in Richtung RADIUS Server auftreten, dem für eine Antwort die übrigen drei Nachrichten zur Verfügung stehen. Das Authenticator Feld wird herangezogen, um die Antwort des RADIUS Servers authentifizieren zu können. Dabei unterscheidet man zwischen

- Request-Authenticator, der in Requests vorkommt und eine 16 Byte Zufallszahl beinhaltet
- Response-Authenticator, vorkommend bei Accept, Reject und Challenge Nachrichten. Er ist mit einem 16 Byte MD5 Hash gefüllt, der über das gesamte Paket und das geteilte Geheimnis[1] gebildet wird, also

MD5 (Code | Identifier | Length | Request-Authenticator | Attribute(s) | secret).

Im Zusammenhang der RADIUS Erweiterung für EAP sind zwei Attribute hinzugekommen, EAP-Message (Typ 79) und Message Authenticator (Typ 80). In Typ 79 Attributen werden EAP Pakete transportiert. Ist das EAP Paket größer als 253 Byte, wird es über mehrere EAP-Message Attribute verteilt. Die Gegenseite erkennt am Vorhandensein mehrerer solcher Attribute ein fragmentiertes EAP Paket und konkateniert

[1] Der Authenticator (z.B. Access-Point) und RADIUS Server teilen ein Geheimnis, das für eine Kommunikation zwischen beiden Parteien Voraussetzung ist und worauf sich das Vertrauensverhältnis gründet. Diese Zeichenkette ist es auch, die man bei der Konfiguration des Access Points für 802.1x/RADIUS hinterlegen muss.

die Fragmente. Dem Fall, dass ein RADIUS Server die neuen Attribute nicht versteht, wird dadurch entgegengewirkt, indem die am Anfang der Authentifizierung vom Client geschickte EAP-Response/Identity Nachricht in ein Username Attribut (Typ 1) verpackt wird. Dadurch wird dem RADIUS Server Gelegenheit gegeben, eine Proxy-Funktion einzunehmen und die Kommunikation an einen anderen mit entsprechenden Fähigkeiten ausgestatteten RADIUS Server weiterzuleiten.

Zum Schutz der Authentizität und Integrität von EAP-Message Attributen ist das Message Authenticator Attribut vorgesehen. Seine Verwendung ist für Access-Requests optional, verpflichtend hingegen für Access-Challenge, Access-Accept und Access-Reject Nachrichten, die EAP-Message Attribute beinhalten. Die Länge dieses Attributs ist die Konstante 18. In allen vier Nachrichten ist das Value-Feld mit dem Hashwert

HMAC-MD5 (Type | Identifier | Length | Request-Authenticator | Attributes)

gefüllt. Der RADIUS Server bezieht also die vom Authenticator stammende Zufallszahl (Request Authenticator) mit in die Prüfsumme ein. Das gemeinsame Geheimnis (shared secret) geht als Schlüssel in den HMAC-MD5 MIC ein. Ist das Message Authenticator Attribut nicht enthalten oder der enthaltene Hash stimmt nicht mit dem eigens berechneten überein, so ist das gesamte RADIUS Paket stillschweigend zu verwerfen.

Wie nun Schlüsselmaterial vom RADIUS Server zum Authenticator verschickt wird, fällt nicht mehr in den Aufgabenbereich von RFC 3579. In diesem Punkt hat man sich auf RADIUS Attribute von Microsoft verständigt, die in RFC 2548 spezifiziert sind. Diese Erweiterungen beinhalten unter anderem das MS-MPPE-Recv-Key Attribut, das im Microsoft Point-to-Point Encryption Protocol (MPPE) für die Zustellung von Schlüsselinformationen zum NAS, vergleichbar mit dem Authenticator, dient. Es kommt nur in Access-Accept Nachrichten vor.

Große Freude scheint sich mit der Wahl dieses Attributs allerdings nie eingestellt zu haben: Im November 2002 hat die EAP Working Group (vgl. [EAP-ML]) erste Bedenken angemeldet, da im Zusammenhang mit diesem Attribut eine Angriffsmöglichkeit für Known Plaintext Angriffe besteht. Das Standardisierungsgremium NIST steht diesem Attribut ebenfalls kritisch gegenüber und sucht nach Alternativen. In RFC 3579 mit Stand September 2003 ist bislang aber kein anderes Verfahren definiert. Es kam ein Abschnitt hinzu, 4.3.4, in dem diese Schwäche identifiziert und als Ausweg ein Schutz von RADIUS durch IPSec ESP vorgeschlagen wird.

Die Entwicklung in dieser Frage sollte nicht aus dem Blickfeld geraten. Wie in Abschnitt 3.5 erklärt wird mit dem MS-MPPE Attribut der wichtigste Schlüssel in der 802.11i Sicherheitsarchitektur transportiert, so dass dessen Integrität und Vertraulichkeit mit größter Sorgfalt sichergestellt werden muss.

3.4 Nachrichtenaustausch einer EAP Authentifizierung

Unabhängig von der konkreten Wahl einer EAP Methode ist der äußere Rahmen der Authentifizierungsphase der gleiche und diese generelle Struktur soll im folgenden analysiert werden. Im Zuge dieser Phase wird auch entsprechendes Schlüsselmaterial generiert, was für den später verwandten Algorithmus zur Absicherung der Kommunikation benötigt wird, z.B. TKIP oder CCMP. Das Rahmenwerk für EAP Schlüssel wird im nächsten Unterabschnitt genauer betrachtet.

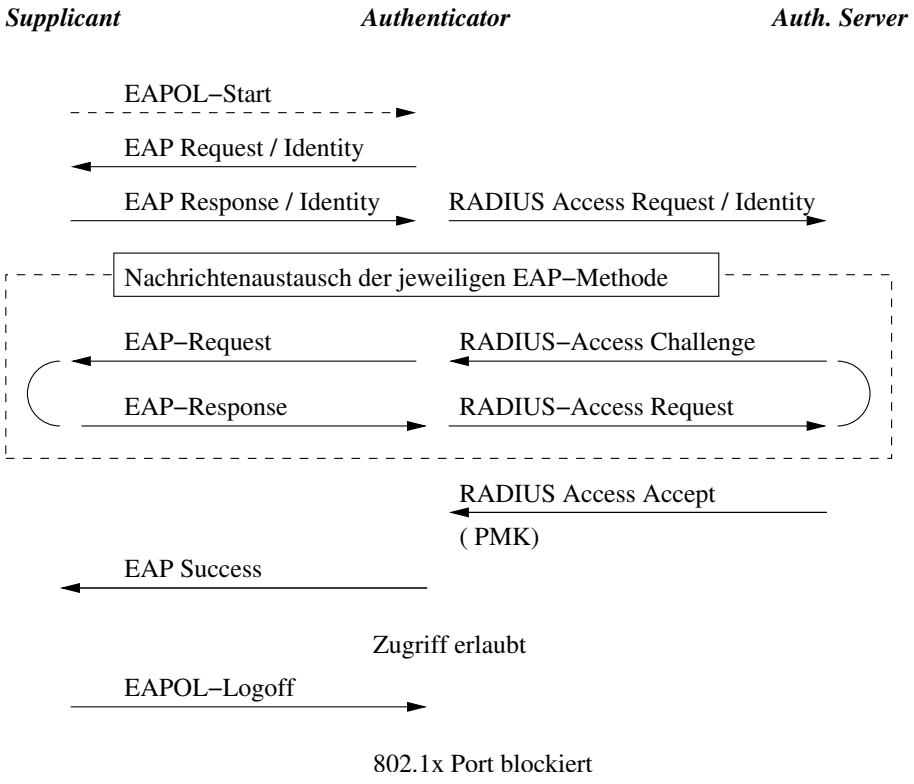


Abbildung 3.14: Nachrichtenaustausch der Authentifizierungsphase

In Abbildung 3.14 ist der Nachrichtenfluss einer EAP-Authentifizierung dargestellt, dem eine erfolgreiche Assoziierung des Clients am Access Point allerdings vorausgegangen sein muss. Zu diesem Zeitpunkt ist der Client bereits im Besitz der Information, ob und wenn ja, welcher Authentifizierungsmechanismus vom Authenticator gefordert wird. Diesbezügliche Angaben sind Bestandteil des RSNIE (Robust Secure Network Information Element), der z.B. in Beacon-Frames mitgesendet wird (siehe auch Anhang C).

1. Die Authentifizierungsphase beginnt durch den Client, der auf ein empfangenes EAP-Request/Identity vom Server mit einem EAP-Response/Identity antwortet und in dieser Nachricht seine Identität[2] mitliefert. Diese Requests werden in regelmäßigen Intervallen vom Authenticator gebroadcastet. Alternativ kann der Client die Authentifizierung auch durch eine EAPOL-Start Nachricht einleiten. Alle Access Points im Einzugsbereich werden diese mit einer EAP-Request/Identity Nachricht beantworten. Im Falle mehrerer Rückmeldungen hat Client den gewünschten auszuwählen.

2. Mit Beginn der EAP-Response/Identity Nachricht reduziert sich die Aufgabe des Authenticators auf das Weiterleiten der EAP Nachrichten, d.h. das Umpacken des EAP Pakets in das jeweilige Trägerprotokoll.

3. Zwischen Client und Authentication Server findet die eigentliche Authentifizierung statt, da nur letzterer im Besitz der Credentials ist. Die Auswahl der EAP Methode fällt in den Aufgabenbereich des Authentication Servers, wobei dessen Wahl für den Authenticator belanglos ist.

[2]Wie mit dieser Identität umzugehen ist, bleibt der EAP Methode überlassen. Im Oktober 2003 wurde hierüber in [EAP-ML] eine Diskussion geführt mit dem Konsens, die EAP-Response/Identity höchstens zu Routingzwecken einzusetzen, ihr ansonsten aber keinen Wert beizumessen. Für eine Beeinflussung des Routings kann der Realm-Teil des Network Access Identifiers (NAI), beispielsweise "@t-online.de" herangezogen werden.

4. Eine vom Client geschickte EAP-Response wird in einen RADIUS Access-Request, eine vom Authentication Server geschickte RADIUS Access-Challenge in einen EAP-Request überführt.

5. Der Ausgang der Authentifizierung wird dem Authenticator durch eine RADIUS Access-Accept bzw. RADIUS Accept-Reject Nachricht mitgeteilt.

6. Abhängig vom Ergebnis wird der 802.1x-kontrollierte Port in den unblockierten Zustand versetzt oder verbleibt im blockierten. Eine gewöhnliche Kommunikation kann zu diesem Zeitpunkt allerdings noch nicht eintreten, da zuvor noch das kurzlebige Schlüsselmaterial zwischen Client und Authenticator verhandelt werden muss (vgl. [802.11i], 5.9).

7. Wünscht der Client die authentifizierte Sitzung zu beenden, wird dies durch Senden einer EAPOL-Logoff Nachricht erreicht.

Das in der RADIUS Access-Accept Nachricht integrierte Schlüsselmaterial wird nun im Authenticator installiert. Der Client hingegen hat dieses Material unabhängig und eigenständig abgeleitet. Es ist nochmals deutlich darauf hinzuweisen, dass obwohl der Authenticator alle Nachrichten der Authentifizierung kennt, dieser daraus keine Rückschlüsse auf Schlüssel ziehen kann. Die anfänglichen Nachrichten einer EAP Authentifizierung sind zwar unverschlüsselt, was aber kein Problem darstellt, da sie keine sicherheitsrelevanten Informationen mitführen. Die in einer EAP Methode stattfindende gegenseitige Authentifizierung samt Austausch vertraulicher Informationen ist allerdings kryptographisch geschützt und somit kann der Authenticator aufgrund fehlenden Schlüsselmaterials keine Kenntnis vom Inhalt der Pakete erlangen.

3.5 EAP Schlüsselmaterial

Am Ende einer erfolgreichen Authentifizierung ist auf beiden Seiten, also Client und RADIUS Server, ein gemeinsamer Schlüssel abgeleitet, der sogenannte "AAA Schlüssel", vom dem ausgehend kurzlebige Sitzungsschlüssel generiert werden. Wie in Abbildung 3.15 erkennbar lassen sich die Schlüssel wie folgt kategorisieren.

- Schlüssel, die nur im Inneren der jeweiligen EAP Methode vorkommen und dort zum Beispiel für Verschlüsselung von Nachrichten (transient encryption keys; TEKs) bzw. als Grundlage für weiteres Schlüsselmaterial dienen (Master Key; MK). Diese Schlüssel werden nicht exportiert.
- In der EAP Methode erzeugte und exportierte Schlüssel. Das sind der Master Session Key (MSK) und der Extended MSK. Beide müssen mindestens 64 Byte als Länge haben. Aus ihnen wird der AAA Schlüssel erzeugt.
- AAA-Schlüssel: Dieser Schlüssel ist es, der sich nach erfolgter Authentifizierung auf Client und RADIUS Server befindet. Letzterer ist für den Transport zum Authenticator zuständig, was mit der RADIUS-Access Accept Nachricht geschieht.

Der AAA Schlüssel wird in der Regel in zwei Hälften geteilt, wovon die ersten 32 Byte als Pairwise Master Key (PMK) bekannt sind (Abbildung 3.16). Aus dem PMK letztlich werden die kurzlebigen Sitzungsschlüssel (transient encryption keys) generiert, wie sie jeweils für TKIP bzw. CCMP benötigt werden. Hiermit befasst sich [802.11i] in aller Ausführlichkeit.

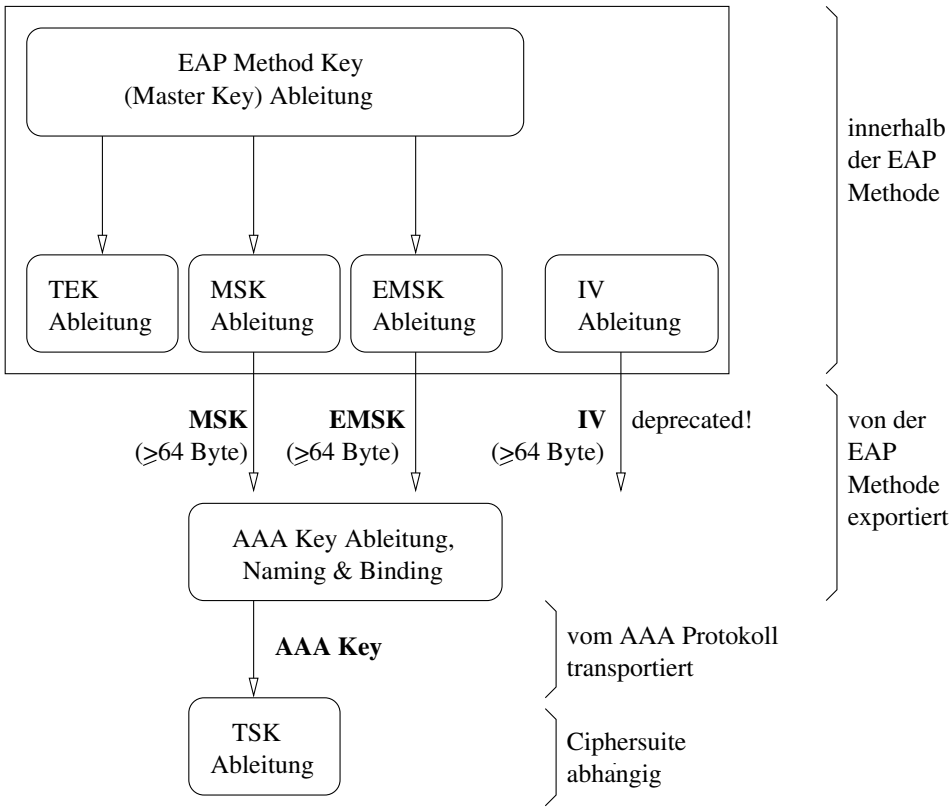


Abbildung 3.15: EAP Schlüssel Framework

Zur Zeit ist die Rolle, die der AAA-Schlüssel im Gesamtwerk einnehmen soll, bzw. dessen Sinn noch in der Diskussion. Im März 2004 gab es in der EAP-Mailinglist eine längere Beitragsserie hinsichtlich der Beziehung zwischen AAA-Schlüssel und dem MSK/EMSK, wobei unter anderem der Vorschlag fällt, den AAA-Schlüssel gänzlich auszusparen, da er in den allermeisten Fällen genau der 64 Byte MSK ist.

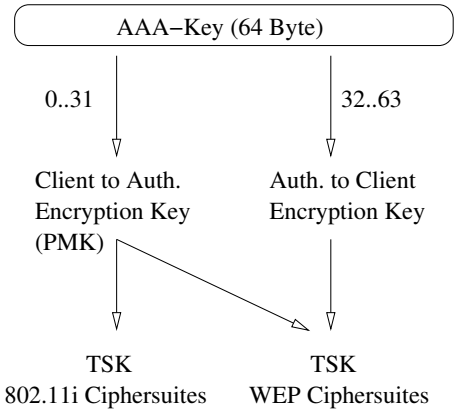


Abbildung 3.16: Transient Session Keys

Wenn überhaupt, dann eignet sich diese Stelle dafür, um auf die in WPA alternative Preshared-Key Lösung einzugehen, die für Endanwender oder kleine Unternehmen angedacht war, die keinen RADIUS Server in Betrieb haben. Hier wird die gesamte Authentifi-

zierungsphase mit 802.1x/EAP umgangen und direkt mit der Generierung des kurzlebigen Schlüsselmaterials begonnen (in Abbildung 3.15 der Kasten "TSK Ableitung"). Dieser Preshared Key ist 32 Byte lang und ist genau der Pairwise Master Key (PMK). Die vom Benutzer im Access Point hinterlegte Passphrase von 8 bis 63 Byte Länge wird durch

$$\text{PMK} = \text{PBKDF2 (passphrase} \,||\, \text{SSID} \,||\, \text{SSIDlength, 4096, 256)}$$

in den 32 Byte PMK überführt. Die PBKDF2 Methode ist Teil von PKCS#5 in Version 2 (Password-based Cryptography Standard). Die Parameter bedeuten, dass 4096 mal iteriert werden soll, um einen 256bit Hash zu generieren.

3.6 Mögliche Schwächen der 802.1x/EAP Architektur

> "Arbaugh's paper shows the necessity of considering the total set of exchanges between the wireless station and the access point – and how we must forget our assumptions about what should happen and instead look for what could happen. What if Packet X is introduced at Point A? What if the authenticator fibs and says an authentication challenge was successful when in fact it was a rogue access point, resulting in the inability to perform the authentication check altogether? The IEEE efforts with which I have been connected have been diligent in producing state machines that show how things are intended to work. But when dealing with security, we must also demonstrate how protocols cope with unintended events, because attackers don't play by the rules."
>
> Robert Moskowitz ([Mo02])

Eine Forschergruppe an der University of Maryland hat im Februar 2002 einen Aufsatz mit dem Titel "An Initial Security Analysis of the IEEE 802.1x Standard" publiziert, in dem Schwächen von 802.1x herausarbeitet und mögliche Angriffe beschrieben werden. Darauf aufbauend sind in der Folgezeit eine Vielzahl an Artikeln erschienen, die dieses Thema weiter ausbauen. Bereits einen Monat nach Erscheinen von [Ar01] hat das Unternehmen Funk Software einen Kommentar zu dem Artikel veröffentlicht, der sich kritisch mit den Vorwürfen auseinandersetzt ([Fu02]). Von Cisco Systems und Agere ist ebenfalls eine Stellungnahme abgegeben worden ([Ag02, Ci02b]). Zusammengefasst beinhaltet der Artikel [Ar01] keine nennenswerten neuen Erkenntnisse, sondern gibt vielmehr eine Zustandsbeschreibung der bis dahin bereits bekannten Schwächen und Sicherheitslücken ab.

Nach der Publizierung hat das Maryland Paper allerdings für Aufsehen gesorgt und für einige Wochen die Diskussion in der 802.11 Mailing List bestimmt. Die beschriebenen Angriffe basierten auf Verwendung eines für WLAN ungeeigneten Authentifizierungsverfahrens, EAP-MD5, welches keine gegenseitige Authentifizierung bietet und sich aufgrunddessen einen Man-in-the-Middle Angriff realisieren lässt. Die weiterhin aufgezeigte Sitzungsübernahme des MitM ist bei WLAN-geeigneten Methoden durch entsprechende Sicherungsmechanismen ebenfalls unmöglich. Dennoch wurde als Konsequenz die Wichtigkeit einer gegenseitigen Authentifizierung noch einmal vor Augen geführt.

Weiterhin in der Diskussion sind Denial-of-Service Angriffe, die übrigens nicht nur auf 802.1x beschränkt sind, sondern auch EAP und selbst 802.11i betreffen. Die Funktionsweise des WLANs kann durch Einstreuen gefälschter

- 802.1x EAPOL-Start und EAPOL-Logoff Nachrichten sowie

- EAP-Success und EAP-Failure Nachrichten

eingeschränkt werden. Das liegt daran, dass diese Nachrichten keinen Schutz der Authentizität und Integrität bieten. Insbesondere das Fluten des Access Points mit EAPOL-Start und die Hinderung des Clients am Beenden des Authentifizierungsvorgangs durch EAPOL-Logoff Fälschung sind hervorzuheben. EAP-Methoden wie PEAPv2 oder EAP-FAST verlassen sich für die Mitteilung des Ausgangs der Authentifizierung nicht mehr allein auf EAP-Success bzw. Failure Nachrichten, sondern führen den Mechanismus der abgesicherten Beendigung ein.

Intensive Diskussionen sind um dieses Thema geführt worden, unter anderem wurde der Vorschlag unterbreitet, den EAPOL Frames ein Authenticator Attribut hinzuzufügen, ähnlich dem HMAC-MD5 Attribut in RADIUS Nachrichten, um die Authentizität und Datenintegrität zu gewährleisten. Alle diesbezüglich gemachten Vorschläge (s. insb. [Ab02]) wurden von der IEEE nicht angenommen. Die durch den Einsatz von 802.1x in Shared Media LANs eingeführten Gefahren müssen, wie in der Spezifikation zu lesen ist, von anderen Mechanismen außerhalb 802.1x abgefangen werden.

> 802.1x takes no steps to provide either reliable authentication or data confidentiality in a shared medium environment. The use of EAPOL in a shared medium environment renders Port-based network access control highly vulnerable to attack. In addition, use of EAPOL in an environment where reliable transmission is not supported will result in frequent timeouts. ([802.1x-9], 7.9)

Ein weiterhin großes Problem sind ungeschützte 802.11 Management (und Control) Frames, wovon sich für DoS-Attacken insbesondere Associate, Re- und Disassociate und Deauthenticate Management Frames eignen. Da sie keine Möglichkeit zur Überprüfung der Authentizität bieten, kann ein Angreifer diese nach Belieben einspielen und so den Betrieb stören. In der Bawug Wireless Mailinglist ([BW-ML]) gab es im April 2003 eine Diskussion zu diesem Thema. Obwohl diese fast ein Jahr zurückliegt, haben die Aussagen weiterhin Gültigkeit. William Arbaugh bemerkt hierzu "Several of us raised DoS issues to TGi and the group was not interested in preventing them." und auf die Nachfrage nach dem Grund "Long story short.....vendors aren't interested. They feel that DoS attacks can come from the RF space just as easily therefore protection at the protocol level is over kill. Personally, I think this is flawed logic."

Weitere von anderen Diskussionsteilnehmern angeführte Gründe sind, dass Authentifizierung dafür in der Reihenfolge vor der Assoziierung stattfinden müsse, und nicht nur in umgekehrter Reihenfolge. Beide Varianten zu unterstützen würde die Zustandmaschine von 802.11i gewaltig verkomplizieren. In diesem Zusammenhang ist auf die Forschung [Di04] zu veweisen. Somit ist es wichtig zu erkennen, dass 802.11 WLAN in missionskritischen hochverfügbaren Anwendungen keinen Platz hat und andere Technologien heranzuziehen sind.

Es gibt eine Reihe weiterer Schwachpunkte im Protokoll, die bekannt sind, wobei in diesem Rahmen nur ein Kritikpunkt exemplarisch angeführt werden soll, und zwar die Gegenmaßnahmen (countermeasures) bei Fehlschlagen der TKIP Integritätsüberprüfung eines Datenpakets ([802.11i], 8.3.2.3.2). Ob sich dieser Mechanismus im endgültigen Standard wiederfindet, bleibt abzuwarten.

Bei Fehlschlagen der Überprüfung des Michael MIC bei einer Unicast- wie Multicast-Nachricht sind folgende Schritte einzuleiten:

1. Der Zwischenfall ist aufzuzeichnen und der Systemoperator zu benachrichtigen.

2. Alle empfangenen Pakete sollen verworfen und das Versenden von Paketen soll für 60 Sekunden eingestellt werden, mit der Ausnahme von 802.1x EAPOL Nachrichten, um neuen Schlüsselaustausch nicht zu unterbinden.

3. Tritt innerhalb von 60 Sekunden eine zweite fehlerhafte MIC Prüfsumme auf, ist für 60 Sekunden jegliche Kommunikation einzustellen.

4. Danach wird zur Einrichtung neuen kurzlebigen Schlüsselmaterials (pairwise keys) der 4-Way Handshake eingeleitet.

Ein Angreifer würde also erfolgreich den Betrieb zum Stillstand bringen, wenn er alle 59 Sekunden eine verfälschte Nachricht in das Netz einstreut. Zum Glück ist dies nicht so einfach. Der TSC (TKIP Sequenzzähler) muss richtig gewählt sein, ist er außer der Reihe, wird das verschlüsselte Paket sowieso verworfen. In [Ho03] wird eine zum Erfolg führende Vorgehensweise vorgedacht. Damit eine kryptographische DoS-Attacke glücken kann, muss ein gültiger Frame während der Übermittlung abgefangen sowie eine Zustellung zum legitimierten Empfänger verhindert werden. Nun ist die MIC entsprechend zu modifizieren, dass sie als ungültig erkannt wird. Der ICV ist auf diesen fehlerhaften MIC neuzuberechnen, was nicht unmöglich ist, weil die ICV Funktion ein linearer CRC ist. Das hieraus resultierende Paket ist dem Client zuzustellen, und zwar bevor diesen irgendein Paket mit höherem IV erreichen kann. Nur auf diesem Wege kann ein MIC Failure ausgelöst werden.

Kapitel 4

EAP Methoden

Der in Abschnitt 3.4 eingeführte Nachrichtenaustausch beschreibt den äußeren Rahmen einer 802.1x/EAP Authentifizierung. Sie wird vervollständigt, indem anstelle des gestrichelt umrandeten Nachrichtenpaares in Abbildung 3.14 ein konkretes EAP Authentifizierungsverfahren eingesetzt wird. In Kapitel 4 werden nun die wichtigsten EAP Methoden vorgestellt, die einer Verwendung in 802.11 WLAN zugedacht sind. Die dabei gewählte Reihenfolge lehnt sich an das zeitliche Erscheinen des jeweiligen Verfahrens an.

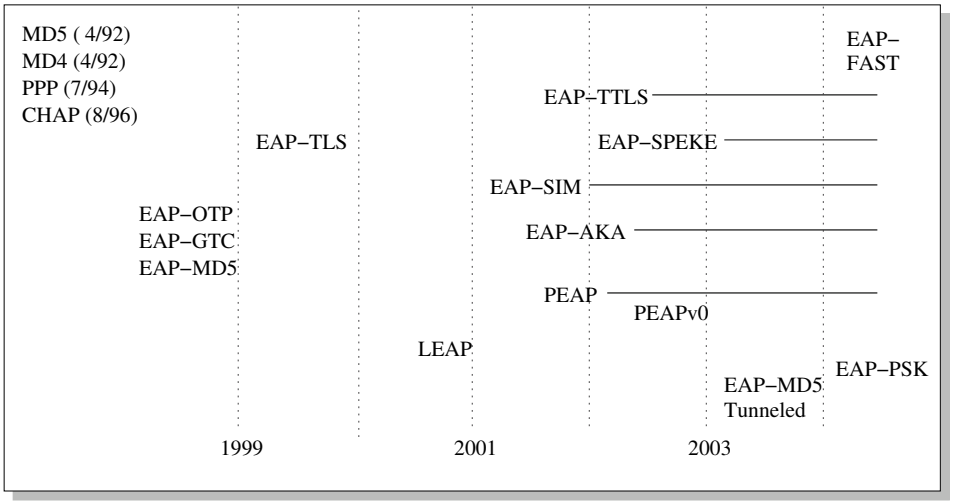


Abbildung 4.1: Zeitliche Entwicklung von EAP-Methoden

Einleitend werden zwei passwortbasierte Verfahren vorgestellt, EAP-MD5 und Cisco LEAP, deren Verwendung in WLAN zwar aus heutiger Sicht nicht mehr zu vertreten ist, die aber für gewisse Zeit ihre Existenzberechtigung hatten. Es folgt EAP-TLS - die bis zum heutigen Tage einzige außerhalb von RFC 2284 standardisierte EAP Methode. Vom sicherheitstechnischen Standpunkt steht einem Einsatz von EAP-TLS nichts entgegen, denn in den fünf Jahren seit Bestehen sind keine gravierenden Sicherheitslücken oder Schwächen im Protokoll entdeckt worden. Vieleher steht einer starken Verbreitung die Komplexität und der administrative Aufwand der zugrundeliegenden Public Key Infrastruktur im Wege.

Diesem Nachteil entgegenwirken sollen die darauffolgend präsentierten "getunnelten" Methoden PEAP, EAP-TTLS und EAP-FAST, die auf EAP-TLS aufbauen. Sie zielen darauf ab, dessen Sicherheit mit Flexibilität sowie Einfachheit in der Anwendung zu vereinen und sind als äußerst zukunftsträchtige Authentifizierungsverfahren einzustufen.

Ihre Entwicklung wird von Microsoft und Cisco sowie der in diesem Marktsegment eine Schlüsselrolle einnehmenden Firma Funk Software vorangetrieben, was auf eine längere Präsenz am Markt schließen lässt.

Dass eine Authentifizierung nicht unbedingt passwort- oder zertifikatbasiert sein muss, soll das nächste beschriebene Verfahren, EAP-SIM, deutlich machen. Dabei wird eine SIM Karte verwendet, wie sie in jedem im GSM Netz operierenden mobilen Endgerät vorhanden ist. Die auf der SIM Karte gespeicherten Informationen dienen dabei zum Beweisen der Identität, so dass sich der Einrichtungsaufwand am Client auf das Verbinden der SIM Karte mit dem Gerät reduziert. Somit eignet sich EAP-SIM insbesondere für Wireless ISPs zur Benutzerauthentifizierung im Hotspot-Umfeld. Auch für UMTS ist eine EAP Methode in der Entwicklung, EAP-AKA, auf die kurz verwiesen wird.

Als letztes wird das vom kryptographischen Gesichtspunkt sehr interessante Verfahren EAP-SPEKE angeschnitten. Es zählt zur Familie der Preshared Keys Methoden, der zur Zeit großes Interesse zukommt. Insbesondere die France Telekom engagiert sich mit EAP-PSK in diesem Bereich. EAP-SPEKE soll in diesem Rahmen als Stellvertreter dienen.

Die Frage, warum es nicht eine, sondern eine kaum überschaubare und ständig wachsende Menge von EAP Methoden gibt, ist durchaus berechtigt, zumal einige sehr große Ähnlichkeiten und gleiche Zielsetzungen aufweisen. Ihr soll zunächst nachgegangen werden, zumal sie auch einfach beantwortet werden kann. Der Grund ist, dass niemand davon abgehalten werden kann, eine EAP Methode zu entwickeln und sie als Internet Draft bei der IETF zu veröffentlichen. In diesem Zuge wird auch bei der IANA eine Typnummer reserviert.

Diese sogenannten Individual Drafts finden nun mal mehr, mal weniger Beachtung. Für viele EAP Methoden ist sich kein hinreichendes Interesse aufkommen und ihre Entwicklung wurde eingestellt (z.B. EAP-MAKE oder EAP-Archie). Eine sehr gute Übersicht über alle reservierten EAP Typen wird in der Anfang April 2004 erschienenen Abhandlung [18] gegeben.

Nach sechs Monaten werden Internet Drafts aus dem Verzeichnis der IETF entfernt, was aber nicht zwangsläufig auf eine Einstellung der Entwicklung schließen lassen darf. Dies ist beispielsweise zur Zeit bei EAP-TTLS der Fall, wobei das Update nur aufgrund größerer Änderungen im Protokoll nicht rechtzeitig erfolgt ist. Die wenigste Begeisterung kommt sicherlich bei Verfahren auf, die auf Patenten beruhen und deren geistiges Eigentum urheberrechtlich geschützt ist (engl. IPR-encumbered methods). Dass diesen Verfahren wenig Beachtung geschenkt wird, liegt in den meisten Fällen aber nicht an mangelnder Qualität, sondern vielmehr an der mit Kosten verbundenen Implementierung, die gemeinhin abschreckt.

Das Internet Draft Verzeichnis dient Arbeitsgruppen als Sammelstelle für Dokumente, die sich zur Zeit in der Entwicklung befinden. Sobald die Spezifikation eines Verfahrens hinreichend stabil ist, ist eine Publizierung an dieser Stelle ratsam, um einer möglichst breiten Öffentlichkeit zur Verfügung stehen. Das hat den Vorteil, dass die Wahrscheinlichkeit steigt, noch rechtzeitig Fehler und sonstige Unzulänglichkeiten aufzudecken. Wichtig ist festzuhalten, dass diese Dokumente in keinerlei Hinsicht offiziellen Status besitzen und für die Entwickler keinerlei Verpflichtungen resultieren. Die meisten derzeit spezifizierten EAP Methoden liegen in Form eines Internet Drafts vor.

Diese Stadium wird durch die Publikation als RFC verlassen. Hierfür ist eine Anfrage

bei der IESG (Internet Engineering Steering Group) einzureichen, die eine Entscheidung innerhalb weniger Wochen fällt. Beispielsweise hat Cisco Systems Anfang April 2004 diesen Weg eingeschlagen und die eigene EAP Methode (EAP-FAST) als Informational RFC angemeldet. Weitere Informationen sind in RFC 2418, IETF Working Group - Guidelines and Procedures, zu finden. Dieses und weitere Stadien sind nämlich zu durchlaufen, bis eine Spezifikation in einen Standard übergeht.

Desweiteren ist hervorzuheben, dass EAP Methoden von keiner Institution oder Arbeitsgruppe betreut werden. Die IEEE ist Schirmherr für 802.11i und die IETF EAP Working Group für EAP sowie damit zusammenhängende Protokolle und Mechanismen. Der Zuständigkeitsbereich der EAP WG umfasst allerdings nicht die Betreuung von EAP Methoden. Das ist auch der Grund, warum EAP-MD5, ein völlig unsicheres Verfahren, auch weiterhin als mandatory-to-implement in der EAP Spezifikation verbleibt, da die EAP WG ein Entfernen als Übertreten ihres Zuständigkeitsbereichs sieht.

EAP Methode	Entwickelt von
EAP-MD5, EAP-OTP, EAP-GTC	RFC 2284
EAP-TLS	Microsoft
PEAP	Microsoft, Cisco Systems
EAP-TTLS	Funk Software
EAP-FAST	Cisco Systems
EAP-SPEKE	Phoenix Technologies
EAP-PSK	France Telecom
EAP-SIM	Nokia, Cisco Systems
EAP-AKA	Nokia, Ericsson

Tabelle 4.1: Entwickler einiger EAP Methoden

So liegt die Initiative schlussendlich bei den Herstellern (vgl. Tabelle 4.1), deren Motivation zur Entwicklung einer EAP Methode aus dem Versprechen eines wirtschaftlichen Vorteils resultiert. Beispielsweise wenden Telefonnetzbetreiber und Handyhersteller ihre Bemühungen auf die Entwicklung von Verfahren, die diese Netze samt ihren Credentials in Anspruch nehmen, oder auch auf sehr leichtgewichtige Methoden, die auch auf rechenschwächeren mobilen Geräten effizient laufen. Konsequenterweise ist in den Geräten bzw. in der Software dieser Hersteller insbesondere "ihr" Verfahren vertreten. Entsprechend kritisch sind somit auch die Whitepapers anzusehen, die von solchen Firmen publiziert werden, z.B. [Mee03, In03, Ph03]. Man wird meistens mit einer verzerrten, die eigene Situation jeweils vorteilhaft bewertenden Darstellung konfrontiert.

Bevor auf konkrete EAP Methoden eingegangen werden kann, sind noch gewisse Grundlagen zu schaffen. Diese bestehen aus der Vorstellung zweier Kriterienkataloge. Zum einen die Aufstellung von Gefahren und Bedrohungen, denen eine EAP Methode im Wireless Umfeld ausgesetzt ist, zum anderen einen daraus hervorgehenden formalen Anforderungskatalog. Letzterer gibt dem Verfahren Vorgaben hinsichtlich des Protokolls und des Designs.

4.1 EAP Gefährdungsmodell

Im Zuge der Erweiterung von EAP für den Einsatz in WLAN und im Internet sind neue Bedrohungssituationen und Angriffsvarianten hinzugekommen, die in [RFC2284-], Kapitel 7 in einer Übersicht zusammengestellt sind. Eine Kenntnis dieser Angriffspunkte untermauert das Verständnis für die im nächsten Abschnitt präsentierten Anforderungen an eine EAP Methode. Mit der Voraussetzung, dass der Angreifer Zugang zur Sicherungsschicht hat, werden sich seine Angriffe auf folgende Ziele richten:

- Aufspüren von Benutzeridentitäten.
- Modifizieren und Fälschen von EAP Paketen.
- Durchführen von Denial-of-Service Angriffen, insbesondere durch Ausnutzen entsprechender Eigenschaften der basierenden Sicherungsschicht, Spoofen von EAP-Success bzw. Failure Paketen, Wiedereinspielen von EAP Paketen oder durch Generieren von Paketen mit überlappenden Identitäten.
- Ermitteln von Passwörtern durch einen offline stattfindenden Wörterbuchangriff.
- Ausführen eines Man-in-the-Middle Angriffs, um den Client zu sich statt zum "echten" Netzwerk verbinden zu lassen.
- Beeinflussung der EAP Verhandlung dahingehend, dass eine schwächere Authentifizierungsmethode herangezogen wird.
- Aufdecken von Schlüsselmaterial durch Ausnutzen von schwachen Schlüsselableitungstechniken innerhalb von EAP Methoden.
- Ausnutzen von schwachen Ciphersuites, die beide Parteien in der EAP Authentifizierung ausgehandelt haben.
- Versuch eines Downgrading Angriffs mit dem Ziel, dass eine ältere Protokollversion mit ihrer schwächeren Ciphersuite verwendet wird.
- Einbringen in den Kommunikationskanal als Authenticator, um zum Beispiel Client und Authentication Server widersprüchliche Nachrichten zuzuspielen.

4.2 Formale Anforderungen an eine EAP Methode

> We have required that the EAP type provide for Mutual Authentication and produce a session secret. We have not determined if implicit is adequate or explicit is required. (SRP is an example of implicit mutual authentication). [...] It is clear that we have to only use EAP types that are resilient to Man-in-the-middle attacks. We all kind of assumed that. So what are the attack models? [...] It will be worthwhile for the IETF to develop a matrix of features of EAP types so that others can be 'informed consumers'.
>
> Robert Moskowitz, Februar 2002 ([EAP-ML])

In einer längeren und durchaus kontrovers geführten Diskussion nach Erscheinen des Artikels [Ar01] wurde der Ruf laut, einen ganzheitlichen Anforderungskatalog für eine im Wireless LAN Umfeld eingesetzte EAP Methode zusammenzustellen. Diese zu berücksichtigenden Punkte sind in [3] in Form eines IETF RFCs für informative Zwecke aufgelistet. Die Anforderungen werden mit RFC-typischer Benennung MUST, SHOULD und MAY in drei Kategorien unterteilt und mit

- "Nichtkonform", wenn alle verpflichtenden Bedingungen nicht erfüllt sind,

- "Bedingt geeignet", wenn alle verpflichtenden, aber nicht alle empfohlenen Anforderungen erfüllt sind,
- "uneingeschränkt geeignet", wenn sowohl alle verpflichtenden wie alle empfohlenen Anforderungen erfüllt sind

bezeichnet. Ausdrücklich werden die in RFC 2284 als verpflichtend zu implementierenden EAP Methoden, nämlich EAP-MD5, EAP-OTP und EAP-GTC, als nicht mehr konform zu diesen Bedingungen angeführt.

Dieses Dokument ist ein wichtiges Verbindungsglied für IEEE 802.11i, IEEE 802.1x und EAP. Wenn überhaupt stellen diese gegenseitig nur sehr allgemeine Forderungen. Beispielsweise stellt die IEEE an das Protokoll zwischen Authenticator und Authentication Server die nur sehr allgemeinen Bedingungen ([802.11i], 5.9.4),

- den beiden Parteien gegenseitige Authentifizierung zu ermöglichen,
- einen Kommunikationskanal für den Transport der Authentifizierungsnachrichten vorzusehen sowie
- die Fähigkeit mitzubringen, das vom Authentication Server generierte Schlüsselmaterial zum Authenticator zu übertragen. Hierbei ist die Authentizität des Absenders, die Integrität des Schlüsseltransfers sowie die Vertraulichkeit des Schlüssels gegenüber allen anderen Teilnehmern sicherzustellen.

Als geeignete Protokolle hierfür werden RADIUS und Diameter vorgeschlagen.

4.2.1 Verpflichtende Eigenschaften

Erzeugen von Schlüsselmaterial (Generation of symmetric keying material). Aus der Spezifikation der EAP Methode muss deutlich die Existenz und Ableitung des zu exportierenden Schlüsselmaterials, des MSK und EMSK, hervorgehen (jeweils mindestens 64 Byte lang; siehe auch Abschnitt 3.5).

Unterstützung gegenseitiger Authentifizierung (Mutual authentication support). Damit eine EAP Methode überhaupt in WLAN Bestand haben kann, muss gegenseitige Authentifizierung erfolgen. Bei den für PPP definierten Protokollen PAP und CHAP beispielsweise erfolgt nur einseitige Authentifizierung, und zwar des Benutzers, was durchaus ausreichend war, da der Kommunikationskanal (Telefonleitung) ausreichende Sicherheit mitbringt. Unabdingbar für den Einsatz in Shared Media LANs ist aber, dass der Client auch die Echtheit des Servers verifiziert. Durch Hintereinanderausführung zweier in entgegengesetzter Richtung durchgeführten, voneinander unabhängigen Authentifizierungen ist diese Forderung übrigens noch nicht erreicht - die beiden Verfahren müssen (kryptographisch) ineinander verzahnt sein [1].

Synchronisation von Zuständen Die EAP Methode muss Authentizität der Datenherkunft sicherstellen und Schutz gegen unauthorisierte Modifikation von EAP Nachrichten, einschließlich EAP-Requests und Responses, bieten. Auch EAP-Success und Failure Nachrichten zählen hierzu, da sie ungeschützt sind und somit von einem Angreifer problemlos gefälscht werden können; PEAP und EAP-FAST führen den "protected termination" Mechanismus ein, um dieser Forderung gerecht zu werden [2].

[1] Ein gutes Beispiel hierfür ist MS-CHAPv2, auf das im Abschnitt 4.9.3 näher eingegangen wird, wenn auch in einem anderen Zusammenhang.

[2] Die Einstufung dieser Anforderung als "verpflichtend" ist noch nicht als endgültig anzusehen, viele Stimmen sprechen sich für eine der anderen beiden Kategorien aus. Weiterhin ist die unpräzise Titulierung in der Kritik (vgl. [EAP-ML]).

Resistenz gegen Wörterbuchangriffe Insbesondere passwortbasierte Verfahren, die für Anfälligkeit gegen Wörterbuchangriffe bekannt sind, müssen entsprechende Gegenmaßnahmen definieren.

Schutz gegen Man-In-The-Middle Angriffe Diese Angriffe können durch kryptographische Bindung (vgl. Abschnitt 4.7.5), Integritätssicherung, Wiedereinspielungsschutz und Unabhängigkeit der Sitzung erreicht werden. Letzteres bedeutet, dass aus passiv erfolgter Aufzeichnung einer Sitzung in Kombination mit aktiven Angriffen (z.B. Kompromittierung des MSK und EMSK) keine Kompromittierung von zurückliegenden oder zukünftigen MSKs oder EMSKs ermöglicht wird.

Kryptographisch geschützte Verhandlung von Ciphersuites (Protected ciphersuite negotiation). Eine EAP Methode muss die Fähigkeit mitbringen, über die Ciphersuite abzustimmen, die für den Schutz des EAP Nachrichtenaustauschs verwendet wird. Weiterhin muss die Integrität dieser Verhandlung geschützt sein.

Schlüsselstärke (Key strength). Durch Abschätzung der effektiven Schlüsselstärke soll die Entscheidungsfindung eines Anwenders, ob das Verfahren in der beabsichtigten Einsatzumgebung überhaupt geeignet ist, vereinfacht werden. Ist die effektive Schlüsselstärke (in der Einheit Bit) N, dann bedeutet das, dass das beste zur Zeit bekannte Verfahren, den Schlüssel aufzudecken, mit hoher Wahrscheinlichkeit einen durchschnittlichen Aufwand von 2^{N-1} Operationen (in der Schwierigkeit einer typischen Blockchiffre-Operation) erfordert.

4.2.2 Empfohlene Eigenschaften

Fragmentierung Die Aufgabe der Fragmentierung und des Wiederzusammensetzens fällt in den Zuständigkeitsbereich der jeweiligen EAP Methode und sollte bei erwarteten Paketen größer als 1020 Byte, der minimalen MTU (maximum transfer unit) eines EAP Pakets, implementiert werden. Beispielsweise kann der Transport eines Zertifikats eine bis zu 16 MB große Nachricht bedeuten (vgl. z.B. [7], 2.4).

4.2.3 Sinnvolle Eigenschaften

Channel Binding Da der Authenticator bei einer Authentifizierung lediglich für die Durchleitung der Nachrichten zuständig ist, der eigentliche Nachrichtenaustausch aber zwischen Client und Authentication Server stattfindet, kann der Client demzufolge auch nicht die Identität des Authenticators verfizieren. Das Vertrauen des Clients gegenüber dem Authenticator basiert allein auf der Annahme, dass zwischen Authenticator und Authentication Server das geforderte Vertrauensverhältnis besteht und gesichert ist. Unter Channel Bindung versteht man nun Mechanismen, mit deren Hilfe Client und Authentication Server einen Eingriff des Authenticators in die Unterhaltung erkennen können.

Verbergen der Benutzeridentität (End-user identity hiding)
Da der anfängliche Identity Nachrichtenaustausch unverschlüsselt ist, ist ein Client nicht verpflichtet, seine echte Identität in der EAP-Response/Identity Nachricht preiszugeben, sondern sie zu einem späteren Zeitpunkt, z.B. im gesicherten TLS Tunnel, "nachzureichen". Soll die EAP Methode allerdings auch Roaming meistern können, so ist anfänglich eine Bestimmung des geeigneten Authentication Servers vorauszugehen, zu dem die Authentifizierungsnachrichten geroutet werden. Um dies zu erreichen, wird auf den in RFC 2486 definierten Network Access Identifier (NAI) zurückgegriffen, genauer, auf den das Gebiet ausdrückenden, zweiten Teil (realm part)

des NAI. Der NAI entspricht bei einer PPP Authentifizierung der vom einwählenden Client übermittelten BenutzerID, also zum Beispiel 0001928374655@t-online.de oder fred@3com.com. In der EAP-Response/Identity Nachricht sollte in diesen Fällen also @t-online.de bzw. @3com.com geschickt werden.

Schnelles Wiederverbinden (Fast reconnect)
Ein schnelles Wiederverbinden bedeutet, die Verhandlungsergebnisse einer bereits zuvor eingerichteten Sicherheitsassoziation zu nutzen, um damit effizienter oder in einer geringeren Rundenanzahl eine neue Sicherheitsassoziation aufzubauen. Insbesondere in Anwendungen wie Wireless Roaming ist diese Eigenschaft von Vorteil, da sie das Intervall der Unterbrechung in der Konnektivität verringert. Eine erneute, vollständige Authentifizierung, insbesondere einer benutzerinteraktiven, zum Beispiel durch Eingabe von Benutzername / Passwort oder durch biometrische Erkennung, sollte vermieden werden.
Je weniger neue Berechnungen erfolgen und je mehr bereits verhandeltes Schlüsselmaterial wiederverwendet werden kann, desto geringer ist die Übergangszeit, bis gewöhnlicher Datenverkehr ausgetauscht werden kann. Den Vorgang, wenn ein Client den Einzugsbereich eines Access Points verlässt und in einen neuen eintritt, heißt in Englisch "handover", Übergeben oder Herüberreichen, Weiterreichen (Abbildung 4.2). So findet man an manchen Stellen gleichbedeutend zu "fast reconnect" auch die Forderung "fast handover" oder "fast handoff".

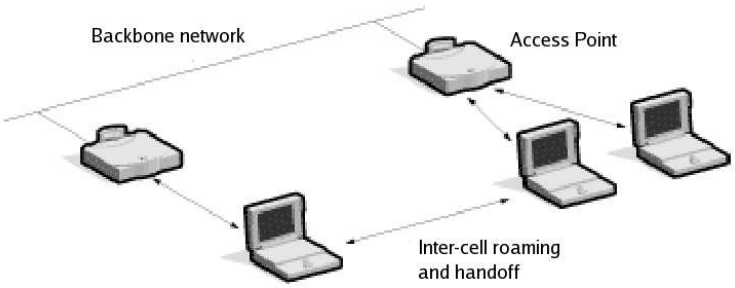


Abbildung 4.2: Fast handoff

EAP Methoden, die Anwendung im WLAN Umfeld finden sollen, werden also sehr genau auf dieses Kriterium überprüft. Schnelles Session Resuming ist bei Anwendungsfällen, die auf Quality of Service (QoS) bauen, wie beispielsweise Voice-over-IP, Grundvoraussetzung. Latenzzeiten von 30ms werden in diesem Bereich als Grenze des Zumutbaren angesehen. Bewegt sich der Client längere Zeit auf der "Grenze" zwischen zwei Zuständigkeitsbereichen von Access Points, was in ein alternierendes An- und Abmelden resultiert, ist diese Latenzzeit natürlich bei weitem zu hoch.

In der Tat ist das von EAP Methoden zur Verfügung gestellte Session Resuming auch nicht für diesen Anwendungsfall ausgelegt, so dass der Blick auf Verfahren außerhalb von EAP zu richten ist. Zur Zeit wird intensiv an einem Protokoll gearbeitet, das schnellen Handover der authentifizierten Sitzung zwischen Access Points bietet. Die IEEE Arbeitsgruppe 802.11f sowie die IETF SEAMOBY WG befassen sich mit der Entwicklung des Inter-Access Point Protokolls (IAPP), welches diese nahtlose ("seamless") Übergabe zwischen 802.11 Access Points möglich machen soll.

4.3 EAP-MD5

EAP-MD5 gehört neben EAP-GTC (Generic Token Card) und EAP-OTP (One-Time Password) zu den drei ersten überhaupt existierenden EAP Methoden und ist in RFC 2284 als verpflichtend zu implementieren vorgegeben. Von der Funktionsweise her ist EAP-MD5 die Umsetzung von PPP CHAP. In der überarbeiteten Version von RFC 2284 ist EAP-MD5 ausdrücklich als nicht mehr konform zu den Anforderungen hervorgehoben, da es beispielsweise weder der Forderung gegenseitiger Authentifizierung noch der Generierung entsprechenden Schlüsselmaterials nachkommt.

Dennoch ist es in den meisten am Markt befindenden Implementierungen vorhanden und wird insbesondere noch in drahtgebundenen Netzwerken zur 802.1x Authentifizierung benutzt. In Abbildung 4.3 ist der Nachrichtenaustausch einer EAP-MD5 Authentifizierung dargestellt. Der Authentication Server sendet einen Challenge an den Client, der den Beweis erbringen soll, in Kenntnis des Passworts zu sein. Dieser berechnet gemäß CHAP-Spezifikation (RFC 1994, 4.1)

MD5 (Identifier | <Passwort> | Challenge Text)

und schickt die resultierenden 16 Byte Hashwert zurück. Der Parameter Identifier ist hierbei das 2 Byte Feld aus dem EAP Paket. Stimmt der empfangene Wert mit dem eigens berechneten überein, so hat der Authentication Server den Client erfolgreich authentifizieren können und erteilt diesem den Zugang zum Netzwerk. Mit der korrekt berechneten Challenge Response beweist der Client, das Passwort zu kennen, ohne es aber direkt über den Link zu schicken. Verglichen mit PAP (Password Authentation Protocol), bei dem Benutzername und Passwort unverschlüsselt übertragen werden, stellt CHAP durchaus eine Verbesserung dar.
Wie dem auch sei, die Verwendung von EAP-MD5 ist im Wireless LAN Umfeld aus Sicherheitsgründen abzulehnen, im Rahmen von 802.11i auch gar nicht möglich, da die Anforderungen nicht erfüllt sind.

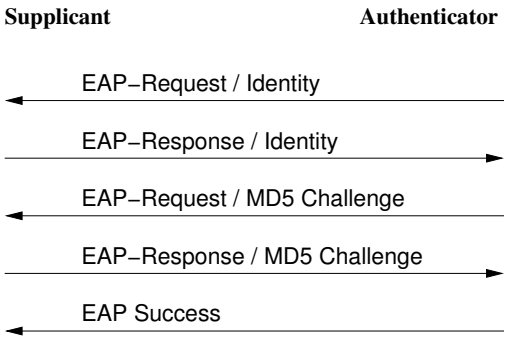


Abbildung 4.3: EAP-MD5

4.4 LEAP

- Can I use a non-Cisco access point to configure LEAP?
- Can I use a non-Cisco client card for LEAP?
- Can I use Microsoft IAS for LEAP?

> The answer to all of the above questions is **no**. [...] All products involved should be Cisco – a Cisco access point, a Cisco RADIUS server and a Cisco client card – to implement LEAP.
>
> Cisco Systems ([Ci03d])

LEAP steht für Lightweight Extensible Authentication Protocol und ist ein proprietäres Authentifizierungprotokoll von Cisco Systems, das im Jahre 2001 entwickelt wurde. Proprietär soll heißen, dass die Spezifikation des Protokolls nicht veröffentlicht und LEAP nur in Cisco-eigenen Produkten bzw. in Produkten von Cisco Lizenzpartnern implementiert ist. Da der Access Point aktiv an der Authentifizierung beteiligt ist[3] und nicht - wie gefordert - transparent ohne Wissen der konkreten EAP Methode als Proxy agiert, zählt LEAP genaugenommen nicht zur Gruppe der EAP Methoden. Aufgrund der breiten Verwendung, insbesondere in größeren WLAN Installationen, sowie der zuletzt erlangten Popularität soll es in diesem Rahmen aber dennoch beschrieben werden.

In einem Cisco Whitepaper ([Ci02a]), das noch heute als Download angeboten wird, wird LEAP die Eigenschaft, als "sicher genug, um in feindlichen Wireless LAN Umgebungen" eingesetzt zu werden, zugesprochen. Diese Überzeugung lässt sich insofern nicht nachvollziehen, als die Unsicherheit von passwortbasierten Verfahren, nämlich die Anfälligkeit gegen Wörterbuch-Angriffe, bereits seit langem bekannt ist, mit EAP-MD5 als prominentem Beispiel.

LEAP baut auf MS-CHAP auf. Die Feststellung, dass es auf MS-CHAPv2 aufbaut, ist gleichrichtig, da beide Protokolle sich in den Punkten, die hier zum Tragen kommen, nicht unterscheiden. Das Passwort wird zum geheimen Schlüssel konvertiert, indem ein zweifacher MD4-Hash über das Klartextpasswort gebildet wird. Weder das Passwort noch der daraus abgeleitete Schlüssel werden - sei es verschlüsselt oder unverschlüsselt - direkt übertragen, womit Cisco die Sicherheit des Verfahrens begründet. Das Schlüsselformat soll dem von Microsoft Windows NT entsprechen und ermöglicht Kompatibilität mit

- Windows NT Domain Services Authentication DB
- Windows 2000 Active Directory databases
- generischen ODBC Treibern mit MS-CHAP Unterstützung

Dabei wird Clients unter Windows angeboten, das Windows Passwort gleichsam für LEAP zu verwenden und beide Login-Vorgänge in einem Schritt zu vereinen.

Diesen Vorteilen steht der Nachteil gegenüber, dass kryptographische Analysen eindeutige Schwächen in den Protokollen von MS-CHAP und MS-CHAPv2 ergeben haben ([Sch98, Sch99, Eis01]), die letzlich auch auf LEAP übertragen wurden. Bevor darauf eingegangen wird, soll das Protokoll im Rahmen der verfügbaren Informationen ([Evol, Thc, Ci01a, Wr03a]) vorgestellt werden.

4.4.1 Protokollbeschreibung

In Abbildung 4.4 ist der Nachrichtenaustausch einer LEAP Authentifizierung gezeigt. Es ist ein typisches Challenge Response Verfahren, das durch Anwendung in beide Richtungen

[3] z.B. [JNw03]: "LEAP makes use of Cisco's vendor-specific attributes (VSAs) to distribute key material. The access point must support the Cisco VSAs and the LEAP algorithm for generating session keys from the key material."

gegenseitige Authentifizierung ermöglicht. Stimmt jeweils die erhaltene Response mit der selbst berechneten überein, ist die Echtheit des Partners festgestellt.

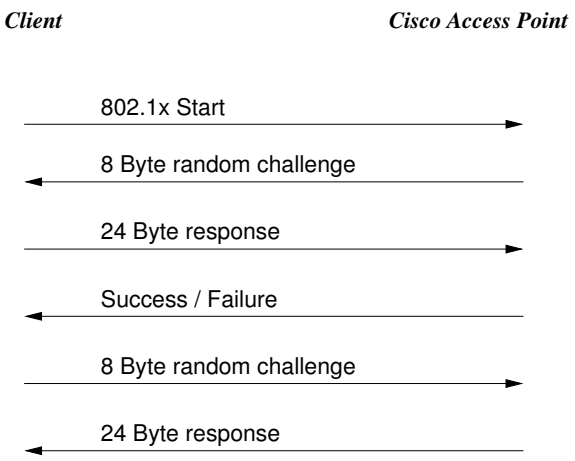


Abbildung 4.4: Cisco LEAP Protokollverlauf

Interessant in diesem Zusammenhang ist ein unveröffentlichtes Whitepaper von Cisco ([Ci01a]), aus dem das ein Jahr später erschienene Whitepaper "A Comprehensive Review of 802.11 Wireless LAN Security" ([Ci02a]) hervorgegangen ist. Ein Abschnitt daraus gibt über die Ableitung des WEP-Schlüssels Aufschluss, der in den nachfolgenden Versionen aber ausgespart wurde.

Es werden die fünf Informationen aus dem Authentifizierungsprozess verwendet, um den 128bit WEP Schlüssel generieren. Der Hashalgorithmus MD5 (RFC 1321) bietet sich insofern an, als er gerade 128bit Hash generiert, die in 104bit WEP Schlüssel und 24bit Initialisierungsvektor aufgehen können:

WEP Key = MD5 (Client NT Key | Challenge vom RADIUS Server | Challenge Response vom Client | Challenge vom Client | Challenge Response vom RADIUS Server)

Da Client wie RADIUS Server über diese Informationen verfügen, kann jede Partei unabhängig von der anderen den Schlüssel ableiten, wobei der RADIUS Server diesen in der RADIUS Accept-Access Nachricht dem Access Point zukommen lässt. Der Rest dieses Abschnitts zeigt auf, wie ein Angreifer anhand der (passiven) Aufzeichnung einer erfolgreich verlaufenden Authentifizierung unter gewissen Umständen das Passwort des Benutzers zurückgewinnen kann. Von diesem Zeitpunkt an kann er sich als legitimierter Benutzer identifizieren und Zugang zum System erlangen.

Zunächst soll die Berechnung der 24 Byte Challenge Response aus der dritten Nachricht aufgezeigt werden.

1. Berechne einen 16 Byte langen MD4 Hash über das Passwort.
2. Verlängere diesen Hash durch Anhängen von fünf Nullen ("00000") auf 21 Byte.
3. Nach dessen Aufteilung in 3*7 Byte lange Fragmente erfolgt eine Verschlüsselung mit DES: Wende den Stromchiffre auf den 8 Byte Challenge mit dem jeweiligen 7 Byte Fragment als Schlüssel an und erhalte drei 8 Byte lange Chiffren.
4. Durch Konkatenation dieser ergibt sich die 24 Byte Challenge Response.

Der NT-Hash soll also bei der DES-Verschlüsselung des Challenges als "Seed" wirken, woraus sich folgende Schwächen ergeben:

- Der NT-Hash verwendet kein Salz, so dass im Vorhinein berechnete Wörterbuchangriffe möglich werden.
- Der DES-Schlüssel für die dritte Challenge Verschlüsselung, H14 H15 00000 ist zu schwach: Es müssen lediglich noch 2 Byte des Schlüssels erraten werden, wofür es $2^{16} = 256^2 = 65536$ Möglichkeiten gibt.

Weiterhin ist als Schwäche von MS-CHAPv2 die Übertragung des Benutzernamens im Klartext anzusehen. So ist es möglich, aus den abgefangenen Paketen einer erfolgreich verlaufenen Authentifizierung die Benutzername / Passwort Kombination zu bestimmen und sich unter dieser Identität am Access Point anzumelden.

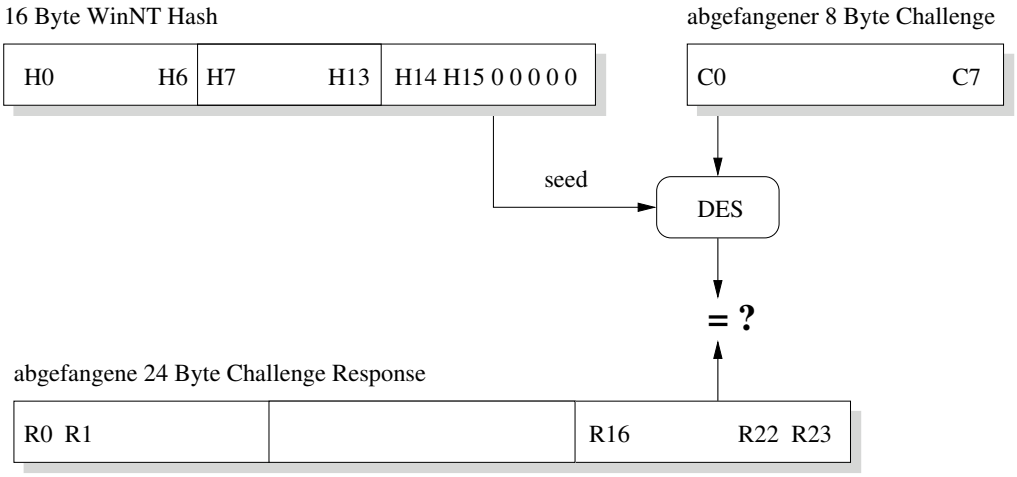


Abbildung 4.5: Weak LEAP

Basierend auf einem Wörterbuch wird als Vorbereitung eine Liste <Passwort, NT-Hash> berechnet. Wird nun ein Challenge und dessen Response aufgefangen, sind die weiteren Schritte:

1. Für jede der 2^{16} Permutationen für $H_{14}H_{15}$ berechne DES (Challenge, seed=$H_{14}H_{15}$00000) und vergleiche auf Übereinstimmung mit den letzten 8 Byte der Response.

2. Die letzten beiden Bytes $H_{14}H_{15}$ sind nun bekannt, so dass gezielt in der NT-Hash Wörterbuch Datei nach den noch in Frage kommenden NT-Hashes gesucht werden kann (brute force search), was eine erhebliche Senkung der Komplexität bedeutet. In [Wr03c] wird die Reduktion des Suchraums exemplarisch von 2.5 Millionen auf etwa 30 angeführt.

Für eine Implementierung bietet sich statt einer Textdatei eine Datenbank (Abb. 4.6) an, mit einem Index Lookup auf H14 H15, was die Geschwindigkeit wesentlich erhöht.

Passwort	WinNT Hash	H14H15 (ind.)
kennwort	B1 05 A8 E2 F3 27 39 A8 24 E4 BE 53 E9 F1 3D B7 00 00 00 00 00	3D B7
12345678	25 97 45 CB 12 3A 52 AA 2E 69 3A AA CC A2 DB 52 00 00 00 00 00	A2 DB
admin	20 9C 61 74 DA 49 0C AE B4 22 F3 FA 5A 7A E6 34 00 00 00 00 00	E6 34

Abbildung 4.6: NT-Hash Datenbank

Angenommen, das Passwort eines Benutzers ist im Wörterbuch des Angreifers enthalten und die von Cisco dokumentierte Ableitung des WEP-Schlüssels entspricht der reellen Implementierung, dann eröffnet sich eine weitere passive Angriffsmöglichkeit.

Der Angreifer ist unter erster Annahme natürlich auch im Besitz des NT-Hashes des Clients. Weiterhin hat er durch Aufzeichnung des Nachrichtenaustauschs die beiden Challenge Response Nachrichtenpaare vorliegen. Das sind alle erforderlichen Parameter, um nach der oben angegebenen Formel den WEP-Schlüssel zu berechnen. Ist der WEP-Schlüssel bekannt, ist natürlich auch der Schlüsselstrom bekannt, da der 24bit Initialisierungsvektor im Klartext gesendet wird. Mit einem gewissen Vorlauf an Nachrichtenaufzeichnung, um die erforderlichen Berechnungen durchzuführen, kann letzlich doch die verschlüsselte Kommunikation zwischen legitimiertem Client und Access Point entschlüsselt werden. Und das sogar zeitgleich.

Da MD5 einen Hash in 128bit Länge produziert, ist ein direktes Übertragen auf die Erfordernisse der WEP Verschlüsselung möglich, und zwar 104bit Schlüssel und 24bit Initialisierungsvektor. Allerdings bleibt noch die Frage zu klären, wie die vier WEP Schlüssel daraus hervorgehen. Der Paketdump im Anhang C zeigt beispielsweise, dass der Access Point in Frame 10 den Unicast Schlüssel mit Index 3 vorgibt.

Nach der eher theorielastigen Beschreibung der Protokollschwäche wird der Blick auf die Implementierung [Thc] gerichtet, die sich durch seine besonders einfache Handhabung auszeichnet. Als Parameter werden - als eine Variante - das (Klartext-)Wörterbuch sowie der Challenge und Challenge Response angegeben. Alle weiteren Vorberechnungen werden automatisch durchgeführt. Auf einen Testvektor angewendet ergibt sich die korrekte Rückgewinnung des Passworts:

```
# ./leap-cracker -f wordlist.txt \
                 -t 5b79dab8bf72ed434ebca8a784466bffb28f6e94280c918d \
                 -c afe811f2ae948bdb

DES1: 5b79dab8bf72ed43
DES2: 4ebca8a784466bff
DES3: b28f6e94280c918d

Matching Password = [blamo]
```

4.4.2 Weak LEAP

Im August 2003 hat J.Wright, Mitarbeiter der Johnson&Wales University, den oben beschriebenen Angriff erfolgreich durchgeführt und ihn der Öffentlichkeit auf der Defcon 11 Konferenz präsentiert. Cisco reagierte daraufhin mit einer knapp gehaltenen Warnmeldung ([Ci03b]), die anscheinend nur geringe Resonanz auslöste, so dass er Anfang Oktober 2003 in einem Beitrag der Bugtraq Mailinglist ein weiteres Mal auf die erheblichen Schwächen von LEAP hinwies. Bereits am darauffolgenden Tag erfolgte, ebenfalls in der Mailinglist, eine Stellungnahme von Cisco ([Ci03e]), in der die Anfälligkeit von LEAP mit der Aussage

“This is not a new attack or new vulnerability of Microsoft MS-CHAP or Cisco LEAP and this proof of concept code demonstrates that simple dictionary based passwords can be deciphered relatively easily.” relativiert wird.

Die Wahl eines starken Passworts würde diesen Angriff unmöglich machen. Zudem verweist Cisco auf ein bereits im Jahre 2001 erschienenes Whitepaper (Vorgänger von [Ci03a]), worin auf die Verwundbarkeit von LEAP eingegangen wird und Gegenmaßnahmen beschrieben sind. Deshalb ist es gerade umso verwunderlicher, dass Cisco in der Stellungnahme um Aufschub der Veröffentlichung des Quellcodes bittet, um genügend Zeit für die Entwicklung von Software-Upgrades zu haben, die im März 2004 abgeschlossen sei ([Ci03b]). Für Außenstehende, aber allen voran für Anwender von Cisco-Produkten dürfte dieser Sachverhalt kaum erklärbar sein.

Durch fortdauernde Negativpresse ist der öffentliche Druck größer geworden, so dass Cisco Anfang Februar 2004 eine neue EAP Methode als LEAP Nachfolger auf den Markt gebracht hat, EAP-FAST. Sie wird in Kapitel 4.9 näher beschrieben.
Zur Vollständigkeit sei angeführt, dass mittlerweile bereits zwei Implementierungen im Internet zur Verfügung stehen ([Evol, Thc]), mit denen ein Wörterbuch-Angriff auf LEAP durchgeführt werden kann. Die Veröffentlichung des Quellcodes von J.Wright ist in einer neuerlichen Übereinkunft auf den 1.April 2004 verschoben worden.

4.4.3 Starke Passwörter

Nicht immer sind Fehler im Design oder Protokoll für erfolgreiche Angriffe verantwortlich. Zwar ist LEAP vom kryptographischen Design kein Meisterwerk, doch hat der Erfolg des oben beschriebenen brute-force Wörterbuchangriffs eine andere Ursache, und zwar die Qualität des zugrundeliegenden Passworts. Seit Bestehen von LEAP weist Cisco auf die Wahl eines “starken” Passworts hin, eines Passworts also, das sich durch folgende Eigenschaften auszeichnet: ([Ci03a]) :

- Mindestlänge von 10 Zeichen
- Mischung von Großbuchstaben und Kleinbuchstaben
- Beinhalten mindestens einer Ziffer (0-9) oder eines nicht-alphanumerischen Zeichens (wie !#@&)
- keinen Zusammenhang mit Benutzernamen oder der User-ID
- kein Eintrag aus einem Wörterbuch, weder der eigenen noch einer fremden Sprache
- oder ein zufällig generiertes Passwort.

Cisco bietet in dem Zusammenhang zwei Beispiele:

- 4yosc10cP!, abgeleitet von "for your own safety choose 10 character Password!"
- cnw84FriDAY, mit"cannot wait for Friday" als gedankliche Hilfe.

Mitunter werden solche kryptischen Zeichenketten auch als ”Shreaker” bezeichnet. Durch das Etablieren einer starken Passwortpolitik im Unternehmen und einer Sensibilisierung der Benutzer, Passwörter nach obigem Muster zu verwenden, biete LEAP nach Firmenangaben weiterhin genügend Sicherheit, um in ”hostile wireless LAN environments” ([Ci02a]) eingesetzt zu werden. Voraussetzung hierfür ist natürlich noch vorhandene Bereitschaft von Seiten der Anwender.

4.5 EAP-TLS

EAP-TLS wurde 1999 von Microsoft entwickelt und ist bis zum heutigen Tage die einzige von der IETF standardisierte EAP Methode. Es bietet unter anderem gegenseitige Authentifizierung, die Möglichkeit zum Schlüsselaustausch, unterstützt Fragmentierung und schnellen Reconnect. Tatsächlich werden alle verpflichtenden sowie die meisten anderen Anforderungen aus Abschnitt 4.4 erfüllt, wodurch einem Einsatz von EAP-TLS in WLAN Umfeld nichts entgegensteht. Der berechtigten Frage, warum sich nicht eine breite Zufriedenheit mit diesem Verfahren einstellen mag, soll ebenso nachgegangen werden, wie einen Blick "hinter die Kulissen" zu werfen und Einzelheiten herauszuarbeiten.

Wie aus dem Namen bereits hervorgeht, basiert EAP-TLS auf dem Transport Layer Security (TLS) Protokoll, das 1998 aus Version 3 des ursprünglich von Netscape entwickelten SSL Protokolls abgeleitet wurde und in RFC 2246 als IETF Standard spezifiziert ist. Dennoch darf die Bezeichnung nicht in die Irre führen, da die Sicherungsmaßnahmen nicht die Transportschicht, sondern die Sicherungsschicht betreffen.

Die Authentifizierung in EAP-TLS erfolgt durch den Austausch digitaler Zertifikate zwischen Client und Server. Die serverseitige Richtung ist bereits aus anderem Umfeld bekannt, und zwar von gesicherten Internet-Webseiten (z.B. Online-Banking). Wird mit dem Browser eine solche Seite aufgerufen, sendet der Server sein Zertifikat, um seine Identität zu beweisen.

Der größte Nachteil bei EAP-TLS ist der hohe administrative Aufwand für Aufbau und Unterhaltung einer Public Key Infrastruktur (PKI). Während in großen Unternehmen bereits oft eine PKI aus einem anderen Zusammenhang existiert, dürfte dies bei mittleren Unternehmen oder im Endanwender-Bereich kaum der Fall sein. Zudem ergeben sich auch bei prinzipiell bekundeter Bereitschaft hierzu nicht zuletzt wegen Mangel an Wissen und Erfahrung unüberwindbare Hürden.

Als weiterer Nachteil ist der Transport des Clientzertifikats zu diesem zu nennen. Hierfür ist ein sicherer Kommunikationskanal erforderlich, da neben dem Zertifikat auch der private Schlüssel mit zu überliefern ist. Desweiteren bereiten alle auf Zertifikaten basierende Authentifizierungsverfahren Schwierigkeiten hinsichtlich der Verifizierung des Serverzertifikats durch den Client. Wenn dem Client nicht bereits im Vorwege das Serverzertifikat oder das Zertifikat der CA zugestellt wurde, dann ist eine Verifizierung des Servers kaum realisierbar. Einzig durch eine bestehende Internetverbindung könnte Klarheit geschaffen werden. Diese Internetverbindung ist, insbesondere in Hotspotumgebungen, aber erst der Zielzustand, dem eine erfolgreiche Authentifizierung vorausgegangen sein muss. In diesem Falle gibt es für den Client keine Möglichkeit, die Echtheit des Zertifikats und somit die vorgegebene Identität des Verhandlungspartners zu überprüfen. Somit ist es unbedingt notwendig, nach erfolgter EAP-TLS Authentifizierung die Überprüfung des erhaltenen Serverzertifikats nachzuholen.

Es ist anzumerken, dass einzig die Spezifikation von EAP-TTLS auf diesen Umstand in gebotener Deutlichkeit hinweist ([11], Abschnitt 12). Bereits im internetbasierten E-Commerce Umfeld ist die Bereitschaft einer solchen Überprüfung als nicht selbstverständlich hinzunehmen:

> Users are "notoriously bad" about following security guidelines.
> When presented with a dialogue saying "the name in the

certificate is different from the name you requested",
most users will simply continue with the transaction.

Nichtsdestotrotz sind die aufgeführten Punkte keine Schwächen oder Unsicherheiten im Protokoll, sondern sollen lediglich als Hinweis auf mögliche erschwerte Rahmenbedingungen bei zertifikatbasierter Authentifizierung aufgefasst werden.

4.5.1 Protokollbeschreibung

In Abbildung 4.7 ist der Nachrichtenaustausch einer erfolgreich ausgehenden EAP-TLS Authentifizierung dargestellt. Das TLS Protokoll lässt gewisse Freiheiten hinsichtlich der Abfolge der TLS Records, so dass die hier gewählte Reihenfolge nur eine Möglichkeit darstellt. Desweiteren ist die hier erfolgte Gruppierung der TLS Records nicht zwingend - sie können auch über mehr EAP Nachrichten verteilt sein. Im Mittelpunkt ist das aus zwei Nachrichtenpaaren bestehende TLS Handshake Protokoll erkennbar, in welchem Client und Server sich über Protokollversion und kryptographische Algorithmen verständigen, sich gegenseitig authentifizieren und mit Public Key Verschlüsselungstechniken ein gemeinsames Geheimnis generieren.

Das TLS Handshake Protokoll setzt sich aus folgenden Schritten zusammen:

- Austausch von Hello Nachrichten. Darin wird eine Übereinkunft über den zu verwendenden Algorithmus getroffen, Zufallszahlen (Nonces) ausgetauscht und eine Überprüfung auf Wiederaufnahme einer unterbrochenen Sitzung durchgeführt.
- Austausch von kryptographischen Parametern, um ein Premaster Secret zu generieren.
- Austausch von Zertifikaten und kryptographischen Informationen, so dass Client und Server sich gegenseitig authentifizieren können.
- Generierung eines Master Secrets aus Premaster Secret und Nonces.
- Abschließende Überprüfung aller Details des Handshakes, z.B. dass beide Seiten im Besitz des (korrekten) Master Secret, sowie der Unverfälschtheit der im Klartext ausgetauschten Nachrichten.

Die genaue Bedeutung aller Nachrichten kann an dieser Stelle nicht erbracht werden, weil es auf eine Beschreibung von TLS v1.0 hinauslaufen würde, dessen Spezifikation mehr als 80 Seiten umfasst. Vielmehr soll der Blick auf die in diesem Nachrichtenaustausch stattfindende gegenseitige Authentifizierung sowie das abgeleitete Schlüsselmaterial gerichtet sein.

Nach dem Empfang der EAP-Request/Identity Nachricht vom Client schickt der Authenticator ein TLS Start zurück, wonach der TLS Handshake eingeleitet wird. In der Client_Hello Nachricht enthalten ist die TLS Versionsnummer, eine Session ID, eine Zufallszahl und Informationen über kryptographische Fähigkeiten des Clients.
Mit der SessionID erkennt der Server, ob es sich um eine neue Sitzung oder um den Wunsch nach Wiederaufnahme einer vorher etablierten Sitzung handelt. Die Entscheidung über eine Wiederaufnahme liegt allein beim Server. Als Antwort sendet dieser eine Server_Hello Nachricht, sein Zertifikat und gegebenenfalls weitere TLS Records zum Client, so dass als Ergebnis dieser beiden Schritte die 48 Byte langen Zufallszahlen, Client.Random und Server.Random, ausgetauscht und eine Ciphersuite ausgewählt worden sind.

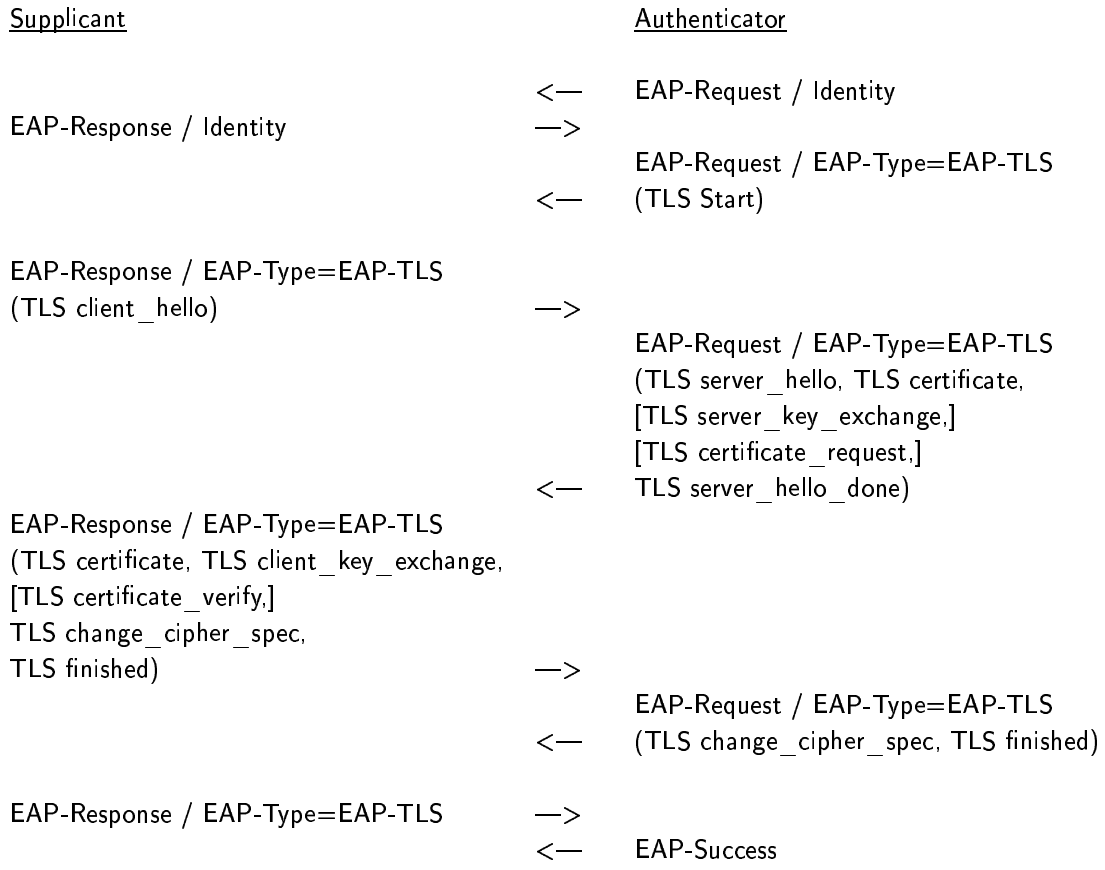


Abbildung 4.7: EAP-TLS Authentifizierung

Der Client generiert ein 48 Byte Premaster Secret, das sich aus 2 Byte Protokollversion und 46 Byte Zufallszahl zusammensetzt. Dieses wird mit dem öffentlichen Schlüssel aus dem Server-Zertifikat verschlüsselt und in der Client_Key_Exchange Nachricht dem Server mitgeteilt.

Die Authentifizierung in der Richtung Client zu Server ist übrigens als optional definiert. Da gegenseitige Authentifizierung aber zu den verpflichtenden Eigenschaften einer EAP-Methode gehört (vgl. 4.4.1), hat EAP-TLS dies auch zu unterstützen, was durch die Certificate_Request Nachricht erreicht wird. Die auf EAP-TLS aufbauenden getunnelten Methoden EAP-TTLS, PEAP und EAP-FAST machen von dieser Option gleichwohl Gebrauch.

Beide Parteien können nun unabhängig voneinander das 48 Byte Master Secret durch

Master Secret = PRF(premaster secret, "master secret", Client.Random | Server.Random)

berechnen. Die Pseudozufallszahlen-Funktion PRF ist in RFC 2246 definiert[4]. Sie bildet einen HMAC mit einer Kombination aus MD5 und SHA-1 in beliebiger, gewünschter

[4] Seite 12: "The secret is partitioned into two halves (with the possibility of one shared byte) as described above, S1 taking the first L_S1 bytes and S2 the last L_S2 bytes. The PRF is then defined as the result of mixing the two pseudorandom streams by exclusive-or'ing them together.

PRF(secret, label, seed) = P_MD5(S1, label + seed) XOR P_SHA-1(S2, label + seed) "

Länge. Als Eingabe benötigt sie (in dieser Reihenfolge) ein Secret, einen Bezeichner und ein Seed.

Das 128 Byte Master Secret ist nötig, um Certificate_Verify und Finished Nachrichten zu erstellen sowie Sitzungsschlüssel für Verschlüsselung und Authentifizierung abzuleiten. Durch Senden einer korrekten Finished Nachricht zeigt der Server, dass er den zu seinem Zertifikat gehörenden privaten Schlüssel kennt. Die Finished Nachrichten werden immer direkt nach Change_Cipher_Spec Nachrichten mitgeschickt, um zu verifizieren, dass der Schlüsselaustausch- und Authentifizierungsprozess erfolgreich waren. Die Finished Nachricht ist die erste mit den zuvor verhandelten Algorithmen, Schlüsseln und Secrets verschlüsselte Nachricht.

Sie beinhaltet jeweilig einen 12 Byte Hash

PRF (master secret, “client finished”, MD5 (handshake messages) | SHA-1 (handshake messages))

PRF (master secret, “server finished”, MD5 (handshake messages) | SHA-1 (handshake messages))

Der Empfänger prüft nun den Inhalt der Finished Nachricht auf Korrektheit, d.h. er berechnet den Hash selbst und vergleicht ihn mit dem empfangenen.

4.5.2 Schlüsselableitung

Bisher ist das innerhalb von EAP-TLS verbleibende Schlüsselmaterial beschrieben worden. Wie in Abschnitt 3.5 allerdings gefordert, muss gewisses Material aus der EAP Methode exportiert werden.

Bezeichne abkürzend random = Client.Random | Server.Random. Die PRF wird erneut angewandt, um sechs jeweils 32 Byte lange Schlüssel zu generieren (Abbildung 4.8). Aus dem ersten, 128 Byte langen String werden Client und Server Encryption Keys und Authentication Keys, aus dem zweiten String Client und Server Initialisierungsvektoren gewonnen. Wie diese nun auf das EAP Schlüssel Framework (vgl. Abbildung 3.15 auf Seite 29) bezogen werden können, ist ebenfalls in Abbildung 4.8 dargestellt, in einer globaleren Sicht in Abbildung 4.9.

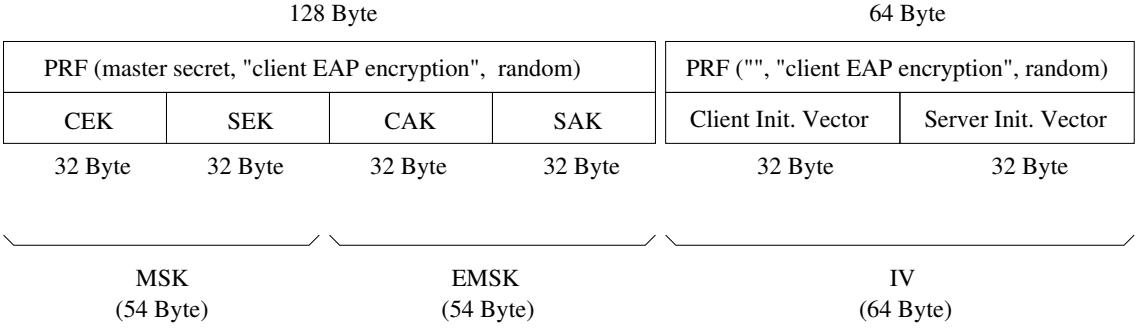


Abbildung 4.8: EAP-TLS Schlüsselmaterial

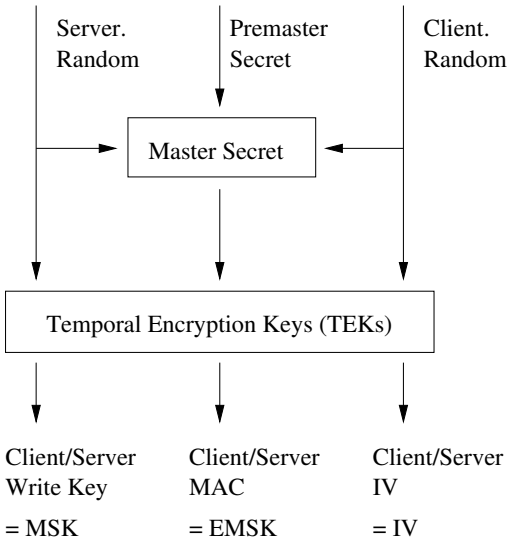


Abbildung 4.9: EAP-TLS Ableitung des Schlüsselmaterials

In der Spezifikation von EAP-TLS sind weiterhin die "Sonderfälle" besprochen, und zwar

- Fehlschlagende Authentifizierung (in beiden Richtungen, also Client kann Server nicht authentifizieren und Server kann Client nicht authentifizieren).
- Verhalten bei notwendig werdender Fragmentierung.
- Wiederaufnahme einer Sitzung (Erfolgreicher Verlauf sowie Fehlschlagen).

Der Aspekt der Fragmentierung wird weder hier noch bei den folgenden EAP Methoden beleuchtet, da die Verfahrensweise relativ mechanisch ist und keine Besonderheiten birgt. Ebenso kann ihm Rahmen dieser Arbeit nicht auf das Verhalten einer EAP Methode im Fehlerfall eingegangen werden.

Vielmehr gilt das Interesse dem Nachrichtenaustausch einer erfolgreich verlaufenen Authentifizierung, sowie dem an den verschiedenen Stellen abgeleiteten Schlüsselmaterial. Zudem ist das Konzept für die Wiederaufnahme einer authentifizierten Sitzung nicht unbedeutend (vgl. 4.2.3) und wird teilweise auch in die Betrachtung mit aufgenommen.

4.5.3 Session Resuming

Der Performance-Flaschenhals bei TLS ist die Schwergewichtigkeit der kryptographischen Public Key Operationen. Auf Seiten des Clients ist dabei vor allem die Verschlüsselung des 48 Byte Premaster Secret mit dem öffentlichen Schlüssel des Servers zu nennen. Für die Wiederaufnahme einer Sitzung (Session Resumption) bietet TLS einen verkürzten Handshake an, der auf Wiederverwendung zuvor verhandelter Parameter aufbaut. Die Leichtgewichtigkeit dieses Mechanismus ist gut zu erkennen: Es erfolgt keine erneute Verhandlung über Algorithmen und Ciphersuites sowie kein erneuter Austausch der Zertifikate. Vielmehr wird das zuvor generierte Master Secret aus dem Sitzungscache herangezogen und direkt mit der Generierung des zu exportierenden Schlüsselmaterials fortgefahren (vgl. Abbildung 4.9).

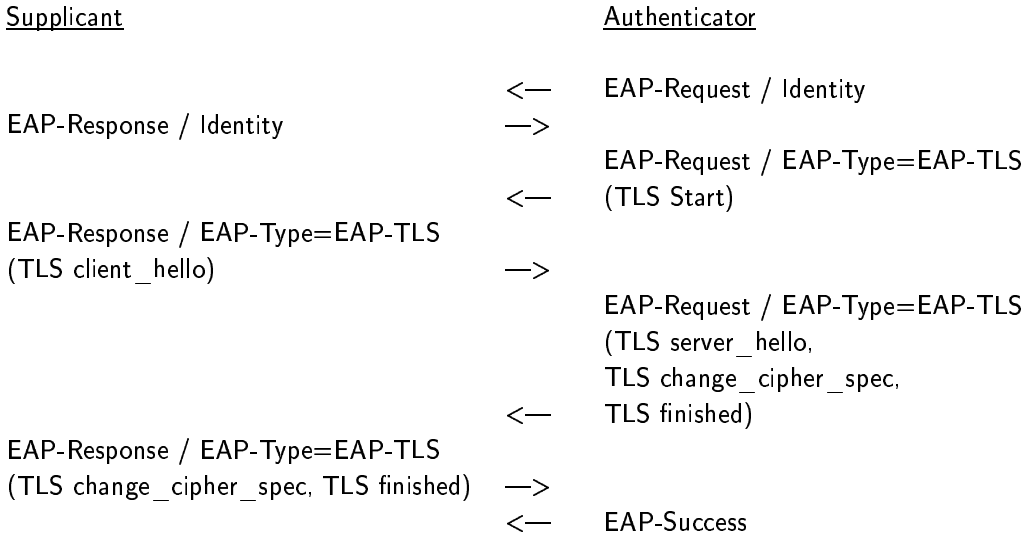


Abbildung 4.10: EAP-TLS Session Resuming

Entscheidend für einen erfolgreichen Ausgang ist der Austausch von Change_Cipher_Spec und Finished Nachrichten und deren erfolgreiche Überprüfung auf Korrektheit. In der Tat wird sich zeigen, dass alle auf EAP-TLS basierenden EAP Methoden das TLS Session Resuming Protokoll heranziehen.

4.6 Grundlagen: Getunnelte EAP Methoden

Um den Aufwand des Aufbaus einer Public Key Infrastruktur zu umgehen aber dennoch die Sicherheit von TLS zu bieten, sind getunnelte EAP Methoden entwickelt worden. Sie zeichnen sich dadurch aus, dass lediglich der Server mit einem Zertifikat ausgestattet und somit nur ein serverseitig authentifizierter TLS Tunnel eingerichtet wird. Durch diesen Tunnel findet die Authentifizierung des Clients statt, und zwar durch Ausführung einer "inneren" Authentifizierungsmethode. Die Auswahl beschränkt sich nicht zwangsläufig auf EAP Methoden, prinzipiell sind auch PAP, CHAP oder MS-CHAPv2 möglich.

Als prominentestes Einsatzbeispiel einer solchen Konstruktion, der Kombination von TLS und einem weiteren Authentifizierungsprotokoll, ist der serverseitig authentifizierte TLS Tunnel, durch den eine HTTP Digest Authentifizierung durchgeleitet wird. Genau dieses Verfahren liegt nämlich mit https:// beginnenden Internetseiten zugrunde. Nach erfolgreicher Validierung des Serverzertifikats findet im Tunnel eine Benutzername / Passwort Authentifizierung statt. Diese beiden Parameter lässt der Client dem Server in der Regel über ein HTML Formular zukommen.

Im Oktober 2002 hat eine finnische Forschergruppe eine grundlegende Schwachstelle dieser getunnelten Methoden aufgedeckt ([As01]), nämlich die Möglichkeit eines erfolgreichen Man-in-the-Middle Angriffs. Glücklicherweise ist diese Schwäche mit relativ geringem Aufwand zu beseitigen und in der Regel im Zuge von Draft Updates in die aktuellen EAP Methoden eingeflossen.

Zunächst soll diese Verwundbarkeit im Allgemeinen aufgezeigt werden. Das Prinzip einer getunnelten EAP Methode wird durch Abbildung 4.11 veranschaulicht. Natürlich müssen

auf dem RADIUS Server nicht unbedingt auch die Benutzeridentitäten liegen, denkbar ist auch ein weiterer (Datenbank-)Server im Hintergrund, z.B. LDAP oder Microsoft Active Directory. Wichtig ist, dass der TLS Tunnel zwischen Client und (RADIUS) Server besteht und zwischen letzterem und dem Authenticator, der während des Authentifizierungsvorgangs lediglich als Passthrough Einheit dient, ein Vertrauensverhältnis existiert. Diese Bedingung soll also das Einbringen des Angreifers zwischen diesen beiden Parteien ausschließen.

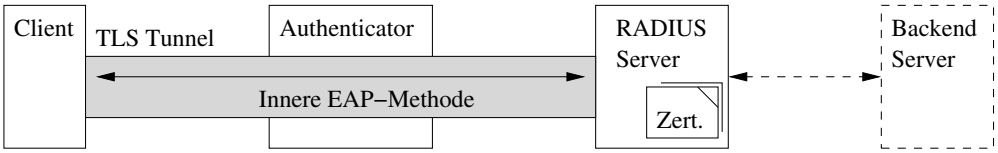


Abbildung 4.11: Getunnelte EAP-Methode

Überraschenderweise entsteht die Angriffsmöglichkeit durch einen Schritt, der eigentlich zur Verbesserung der Sicherheit erfolgt ist, das Bilden eines TLS Tunnels und das Hindurchführen einer inneren beliebigen Authentifizierungsmethode. Auch wenn jede für sich betrachtet ein hohes Maß an Sicherheit bietet, so ist die Komposition unsicher.

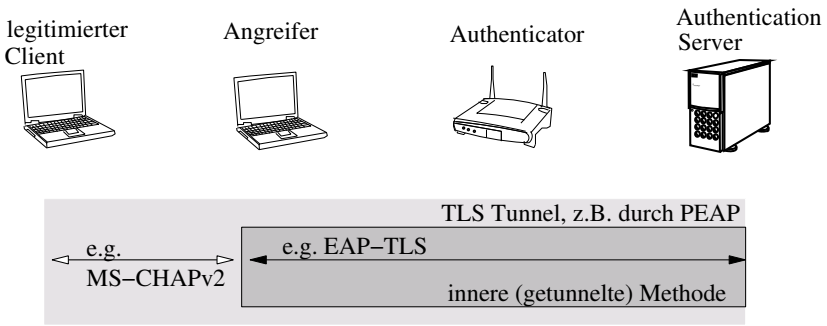


Abbildung 4.12: MitM als Proxy

Es ist möglich, dass ein Angreifer (im folgenden mit MitM bezeichnet) nach Etablierung des TLS Tunnels sich (sinnbildlich) zwischen Client und Authenticator einspielt und als Proxy zwischen Client und RADIUS Server agiert, natürlich alle vertraulichen Informationen herausfilternd. Damit der Angriff gelingt, sind - kurz umrissen - folgende Schritte nötig:

- Der Client hat für die innere Authentifizierung eine ungetunnelte EAP Methode zu wählen. Entweder wird solange gewartet, bis dieser Fall eintritt, oder der MitM bewegt ihn durch eine gefälschte Aufforderung dazu[5]. Diese Beeinflussung der Verhandlung ist als Ziel eines Angriffs erkannt und im EAP Gefährdungsmodell (Abschnitt 4.1) explizit verankert.

[5] Zur Erinnerung: Bevor die eigentliche Authentifizierung stattfindet, beinhaltet jede EAP Methode einen Nachrichtenaustausch, in dem der Server die gewählte EAP Methode bekanntgibt und dem Client die Gelegenheit zum Einspruch gegeben wird. Durch Senden einer Nak Nachricht (Not Acknowledge) kann dieser dem Server signalisieren, dessen vorgeschlagenes Authentifizierungsverfahren nicht zu unterstützen. Entweder verständigen sich beide Seiten auf eine Alternative oder der Server bricht die Authentifizierung ab, falls er kein alternatives Verfahren zulässt.

- Die erste Nachricht vom Client wird abgefangen. Stattdessen beginnt der Angreifer, die innere, und zwar diesmal eine getunnelte, EAP Methode mit dem Server. Ist dieser Tunnel eingerichtet, werden die vom Client geschickten Authentifizierungsnachrichten zum Server weitergeleitet. Die empfangenen Nachrichten werden "umgepackt" und dem Client zugestellt.
- Nach Beendigung der inneren Methode leitet der Angreifer die Sitzungsschlüssel ab. Diese Ableitung basiert auf dem Schlüsselmaterial des äußeren Tunnels.

Es ist offensichtlich, dass es sich hierbei um einen Angriff eher theoretischer Natur handelt und dass Zufall und Glück einen nicht unbedeutenden Anteil am Gelingen ausmachen.

Der Grund für sein Möglichwerden ist, dass der Authenticator sich nicht versichern kann, ob der Client der inneren Methode und der Client, der den Endpunkt des TLS Tunnels bildet, dieselbe Einheit ist, was in einer Kommunikation mit dem "falschen" statt dem "echten" Client münden kann. Weder Client noch Server können verifizieren, dass die das innere Authentifizierungsprotokoll ausführende Einheit gleich dem Endpunkt des äußeren Tunnels ist.

Um diesen Misstand zu beseitigen, muss eine kryptographische Bindung zwischen dem inneren und dem äußeren Protokoll hergestellt werden, beispielsweise durch Einführen eines zusätzlichen Nachrichtenaustauschs, in dem verifiziert wird, dass die Einheiten, die im Besitz des Master Secret sind (aus dem die Sitzungsschlüssel abgeleitet werden), auch die gewünschten sind.

In [As01] wird das Ineinandergreifen zwischen den Schlüsseln im Allgemeinen beschrieben. Letzlich ist aber die jeweilige getunnelte EAP Methode explizit zu betrachten. Im weiteren Verlauf des Artikels wird darauf hingewiesen, dass bei Verwendung von schwachen Authentifizierungsprotokollen innerhalb eines serverauthentifizierten TLS Tunnels der Client unbedingt die Authentizität des Servers sicherstellen muss und, wenn als innere eine getunnelte Methode gewählt wird, diese von der äußeren verschieden sein muss.

Das Konzept der kryptographischen Bindung ist in einem Internet-Draft ([1]) festgehalten. Bei dem nachfolgend beschriebenen PEAP ist dieser Mechanismus ohne größere Abweichung von der Vorgabe mit aufgenommen worden, so dass am Beispiel PEAP dieses Konzept veranschaulicht wird.

4.7 PEAPv2

Protected EAP (PEAP) ist von RSA, Cisco und Microsoft entwickelt worden. Die erste Veröffentlichung fand im August 2001 statt. Im Oktober 2002 hat Microsoft einen Zwischenstand abgezweigt, um ihn im Betriebssystem Windows XP über ein Service Pack zur Verfügung zu stellen. Zur Unterscheidung wird diese Version mit PEAPv0 und die aktuelle Version vom Oktober 2003 mit PEAPv2 bezeichnet. Bei der IANA ist PEAP versionsübergreifend unter der Typnummer 25 geführt. Wenn im weiteren Verlauf von PEAP die Rede ist, so ist das aktuelle PEAPv2 gemeint.

PEAP bietet einen verschlüsselten und authentifizierten TLS Tunnel, durch den eine beliebige EAP Authentifizierungsmethode hindurchgeleitet werden kann. Mit der Draft Version

vom Oktober 2003 ist PEAPv2 gegen alle bisher bekannten Schwächen und Angriffe resistent und erfüllt den gesamten Anforderungskatalog an eine EAP Methode (Abschnitt 4.2). Neben den grundlegenden Eigenschaften, wie der Erzeugung geeigneten Schlüsselmaterials und Unterstützung von Fragmentierung (und Wiederzusammensetzen) sind als Besonderheiten von PEAP anzuführen:

Schutz der Identität Die Identität des Clients ist dadurch geschützt, dass sie erst nach Einrichtung des äußeren TLS Tunnels ausgetauscht und somit nicht im Klartext übertragen wird.

Sichere Beendigung (protected termination). Um dem aus Fälschen von EAP-Success und EAP-Failure resultierenden Denial-of-Service Angriff entgegenzuwirken, ist in PEAP der Mechanismus der "Abgesicherten Beendigung" vorgesehen. Damit stützt sich der Ausgang einer Authentifizierungsverhandlung nicht mehr allein auf das Verschicken dieser beiden EAP Nachrichten.

Flexibilität Innerhalb des Tunnels wird ein generisches Type-Length-Value (TLV) Format zum Nachrichtentransport herangezogen. Dies ermöglicht den Austausch beliebiger Nachrichten zwischen Client und Authentication Server, mit dem Vorteil, jede - auch zukünftige - EAP Methode im Tunnel transportieren zu können, ohne neue individuelle Nachrichtenformate einführen zu müssen.

Kryptographische Bindung Dem in Abschnitt 4.6 geschilderten Man-in-the-Middle Angriff wird entgegengetreten, indem Nachrichten der inneren Methode durch Authentifizierung, Integritätschutz sowie Wiedereinspielung auf Paketbasis einen besonderen Schutz erfahren. Der Forderung der kryptographischen Bindung zwischen den Schlüsseln der inneren und äußeren Methode wurde ebenfalls entgegnet, indem das Schlüsselmaterial der äußeren mit dem Schlüsselmaterial der inneren Methode(n) kombiniert wird, so dass die innere(n) Methode(n) kryptographisch an den TLS Tunnel gebunden ist.

Im weiteren Verlauf dieses Abschnitts wird

- der Nachrichtenaustausch einer erfolgreich verlaufenden Authentifizierung,
- das Konzept der Unterteilung in zwei Phasen
- sowie zwei Besonderheiten beschrieben, und zwar
  - die Aneinanderreihung von mehreren inneren EAP Methoden ("Sequencing") sowie
  - das TLV Format einschließlich Paketbeschreibung und der Umsetzung des "protected termination"-Mechanismus mit Hilfe des Result-TLV.

4.7.1 Sequencing

Die Spezifikation von EAP sieht in einer Authentifizierungsverhandlung keine Ausführung mehrerer Authentifizierungsmethoden vor. Das ist aber bei getunnelten Verfahren auch nach außen nicht der Fall, da diese nach außen den Anschein einer Methode macht. Der Nachrichtenaustausch im Tunnel ist für die Außenwelt nicht durchschaubar, und so ermöglicht PEAP im Tunnel sogar die Ausführung mehrerer EAP Methoden. Dies wird mit "Sequencing" bezeichnet, das in zwei Varianten erfolgen kann (Abbildung 4.13).

- Serielle Authentifizierung. Eine Folge von EAP Methoden wird ausgeführt, nicht zwingenderweise verschiedene. Die erfolgreiche Authentifizierung tritt ein, wenn jede einzelne erfolgreich endet.

- Parallele Authentifizierung. Einleiten einer alternativen EAP Methode nach Fehlschlagen einer oder mehrerer anfänglicher Methoden. Wird irgendeine der Methoden erfolgreich beendet, so wird die gesamte Authentifizierung als "erfolgreich" beendet.

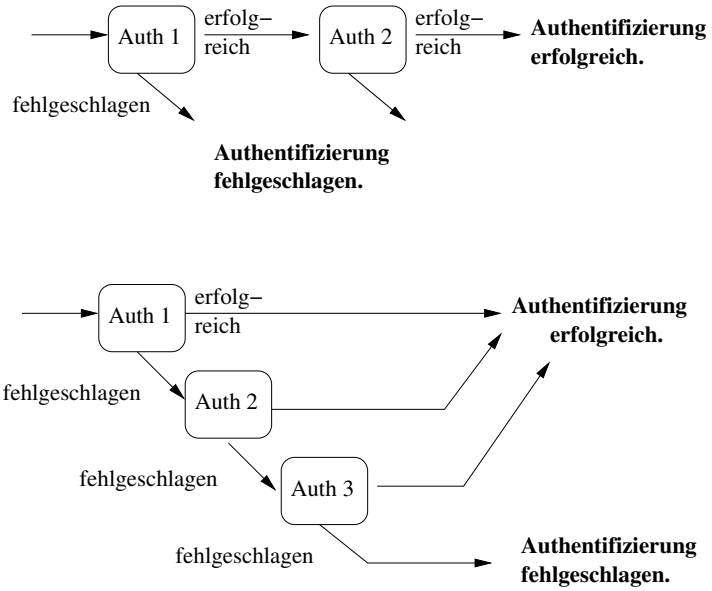


Abbildung 4.13: PEAPv2 Sequencing

4.7.2 Protokollbeschreibung

Die PEAP Authentifizierung besteht aus zwei Phasen. In Phase 1 wird - dem Paradigma einer getunnelten EAP Methode folgend - eine TLS Sitzung initiiert und ein serverauthentifizierter Tunnel aufgebaut. Der dabei generierte Schlüssel, das TLS Master Secret, wird für die Verschlüsselung der Nachrichten in Phase 2 verwendet. Phase 2 kann, muss aber nicht die Ausführung einer oder mehrerer innerer EAP Methoden beinhalten. Die einzige an sie verpflichtende Bedingung ist der Ausgang durch eine abgesicherte Beendigung.

Für den im folgenden beschriebenen Nachrichtenverlauf werden zwei Annahmen getroffen: Zum einen überträgt der Client seine Identität erst im Tunnel, zum anderen ist der Client nicht mit einem Zertifikat ausgestattet. Natürlich unterstützt PEAP auch diesen Fall, aber dann wäre der Vorteil gegenüber EAP-TLS und der Bürde einer PKI aufgehoben.

In Abbildung 4.14 sind die Nachrichten der ersten Phase dargestellt, die zu EAP-TLS sehr ähnlich ist. Denn auch hier findet ein vollständiger TLS Handshake statt. Unterstützt der Client PEAP nicht, so ist der EAP-Request/PEAP Start Nachricht mit EAP-Nak zu antworten.

Der Aufforderung des Servers mit der Certificate Request, den Client zum Schicken dessen Zertifikats zu bewegen, muss dieser - angenommen einmal, er wäre im Besitz eines - nicht nachkommen. Da es nämlich zu diesem Zeitpunkt im Klartext übertragen würde, wäre natürlich seine Identität für jeden anderen in Erfahrung zu bringen. Es gibt eine zweite Gelegenheit für den Server, den Client zum Schicken dessen Zertifikats aufzufordern, und zwar zwischen TLS Finished und der ersten Nachricht von Phase 2, also einer EAP-TLV Nachricht. Auch hier ist der Client nicht verpflichtet, diesem Wunsch nachzukommen.

Durch die Nichtbefolgung wird der Authentifizierungsvorgang übrigens nicht beendet.

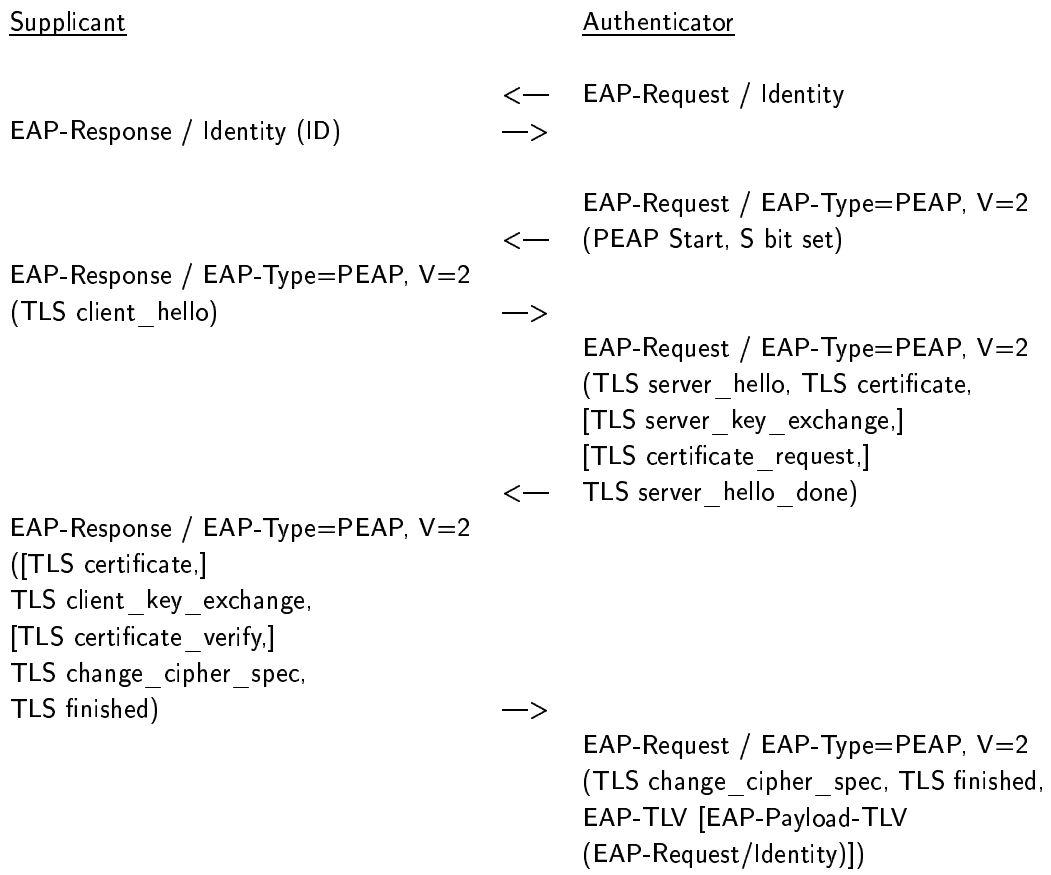


Abbildung 4.14: PEAPv2 (Phase 1)

Die letzte Nachricht in Abbildung 4.14 ist bereits die erste Nachricht von Phase 2. Der TLS Tunnel konnte erfolgreich aufgebaut werden, entweder in Folge eines vollständigen TLS Handshakes bei einer neuen Sitzung oder in Folge einer Wiederaufnahme einer vormals bestehenden Sitzung. Alle Nachrichten in Phase 2 werden durch die verhandelte TLS Ciphersuite geschützt, einschließlich aller EAP Pakete (samt Header) der inneren Authentifizierungsmethode. Ihr Transport durch den TLS Tunnel findet in EAP-TLV Nachrichten statt.

EAP-TLV ist im Oktober 2002 von Lucent, Microsoft und Cisco entwickelt worden und hat im Mai 2003 eine Überarbeitung erfahren. Zwar ist EAP-TLV bei der IANA mit der Typnummer 33 registriert, aber im Grunde genommen nicht mehr als ein Memorandum. Der Draft umfasst acht Seiten und liefert lediglich eine Beschreibung des TLV (Type Length Value) Formats, mit dem Nachrichten im TLS Tunnel zu transportieren sind. Die EAP Spezifikation erlaubt ihren Einsatz auch nur in diesem Zusammenhang, als Transportcontainer für Nachrichten im TLS Tunnel. Auf das TLV Format wird im nächsten Abschnitt genauer eingegangen.

Phase 2 (Abbildung 4.15) beginnt typischerweise mit EAP-Request/Identity Nachricht, woraufhin der Client in der Response seine Identität MyID schickt. Im Gegensatz zu Phase 1 kann sorglos die echte Identität preisgegeben werden, da alle Nachrichten in Phase 2 verschüsselt übertragen werden. Der Server schlägt daraufhin eine innere EAP Methode X vor. Unterstützt der Client das Verfahren nicht, so kann durch EAP-Nak

eine Neuverhandlung eingeleitet werden. Entscheidend ist, dass ein Angreifer auch diese Nachricht nicht zu einem Angriff umfunktionieren kann. Der kryptographische Schutz soll Außenstehende nicht einmal die gewählte innere Methode direkt erkennen lassen, außer infolge statistischer Analyse der Datenpakete.

Prinzipiell muss aber im Tunnel keine Authentifizierung erfolgen. Der Server hat sich möglicherweise schon von MyID überzeugen lassen. Oder er ist in Phase 1 in den Besitz des Client Zertifikats gelangt, dessen Überprüfung erfolgreich war. Wie dem auch sei, EAP Nachrichten einer inneren Methode werden als Payload von EAP-TLV gekapselt (vgl. Abbildung 4.15).

Supplicant		Authenticator
EAP-TLV [EAP-Payload-TLV [EAP-Response/Identity (MyID)]]	—>	
		EAP-TLV [EAP-Payload-TLV
	<—	[EAP-Type=EAP-Request/ EAP-Type=X]]
	— Nachrichtenaustausch der inneren EAP Methode —	
EAP-TLV [EAP-Payload-TLV [EAP-Response/EAP-Type=X or NAK]	—>	
	<—	EAP-TLV [EAP-Payload-TLV [EAP-Request/EAP-Type=X]]
	— Nachrichtenaustausch der inneren EAP Methode beendet —	
EAP-TLV [EAP-Payload-TLV [EAP-Response/ EAP-Type=X]]	—>	
		EAP-TLV [Crypto-Binding-TLV= (Version=0, Received-version=2, Nonce, B1_MAC), Intermediate-Result-TLV(Success),
	<—	Result TLV (Success)]
EAP-TLV [Crypto-Binding-TLV= (Version=0, Received-version=2, Nonce, B2_MAC), Intermediate-Result-TLV (Success), Result TLV (Success)]	—>	
	— Beendigung des TLS Tunnels —	
	<—	EAP-Success

Abbildung 4.15: PEAPv2 (Phase 2)

Zwingend erforderlich ist hingegen eine abgesicherte Terminierung des PEAPv2 Authentifizierungsprozesses, was durch gegenseitiges Schicken von Result-TLV Nachrichten erfolgt.

Durch sie wird das Ende von Phase 2 besiegelt. Ist also die Authentifizierung des Clients erfolgreich verlaufen, so hat ein Austausch

```
<—   EAP-Request / Result TLV: Status = Success
     EAP-Response / Result TLV: Status = Success   —>
```

stattzufinden. Alle anderen Möglichkeiten bedeuten, dass die Authentifizierung fehlgeschlagen ist. Sollte neben der äußeren Methode als innere Methode ebenfalls PEAPv2 gewählt werden, so muss jede einzelne durch diese Nachrichtenfolge terminiert werden. Um dem Man-In-The-Middle Angriff entgegenzutreten, sind Crypto-Binding-TLV vorgesehen, deren genaue Aufgabe in Abschnitt 4.7.5 aufgezeigt wird.

4.7.3 Session Resuming

In Abbildung 4.16 wird die Wiederaufnahme einer vorigen Sitzung gezeigt. Der Server überprüft die Session ID aus der TLS Client_Hello Nachricht mit seinem Sitzungscache. Eine bekannte Session ID deutet ja auf einen Client hin, der eine vorher bestehende, aber unterbrochene Sitzung wiederaufnehmen möchte. Jeglicher Austausch von Zertifikaten sowie die gesamte Phase 2, also die Ausführung innerer EAP Methoden, kann dann ausgespart werden.

Supplicant		Authenticator
	<—	EAP-Request / Identity
EAP-Response / Identity (ID)	—>	
	<—	EAP-Request / EAP-Type=PEAP, V=2 (PEAP Start, S bit set)
EAP-Response / EAP-Type=PEAP, V=2 (TLS client_hello)	—>	
	<—	EAP-Request / EAP-Type=PEAP, V=2 (TLS server_hello, TLS change_cipher_spec TLS finished)
EAP-Response / EAP-Type=PEAP, V=2 TLS change_cipher_spec, TLS finished)	—>	
	<—	EAP-Request / EAP-Type=EAP-TLV Result TLV (Success)

Abbildung 4.16: PEAPv2 Session Resuming

Allem Anschein nach ist dieser Fall optimiert worden, was die Designkriterien von PEAPv2 auch bestätigen ([7], 1):

> Where EAP is used for authentication in wireless networks, the authentication latency is a concern. As a result, it is valuable to be able to do a quick re-authentication on roaming between access points. PEAPv2 supports this capability by leveraging the TLS session resumption facility, and any EAP method running under PEAPv2 can take advantage of it.

Aufgrund ihrer Anfälligkeit gegen MitM Angriffe sind anonyme Ciphersuites nicht erlaubt, was an dieser Stelle aber nicht relevant ist. Cisco's EAP-FAST verfolgt diegleiche Strategie wie PEAPv2, es unterscheiden sich lediglich die TLS Ciphersuites, die für Phase 1 zur Verfügung stehen.

Methode	Verpflichtende Ciphersuite	Key	Auth.	Encryption	Integrität
PEAPv2	`TLS_RSA_WITH_3DES_EDE_CBC_SHA`	168	RSA	3DES EDE CBC	SHA
EAP-FAST	`TLS_RSA_WITH_RC4_128_SHA`	128	RSA	RC4	SHA

In Fachkreisen wird die Performance von PEAPv2 und EAP-FAST im Session Resuming Fall als annähernd gleich bezeichnet.

4.7.4 Nachrichtenformate

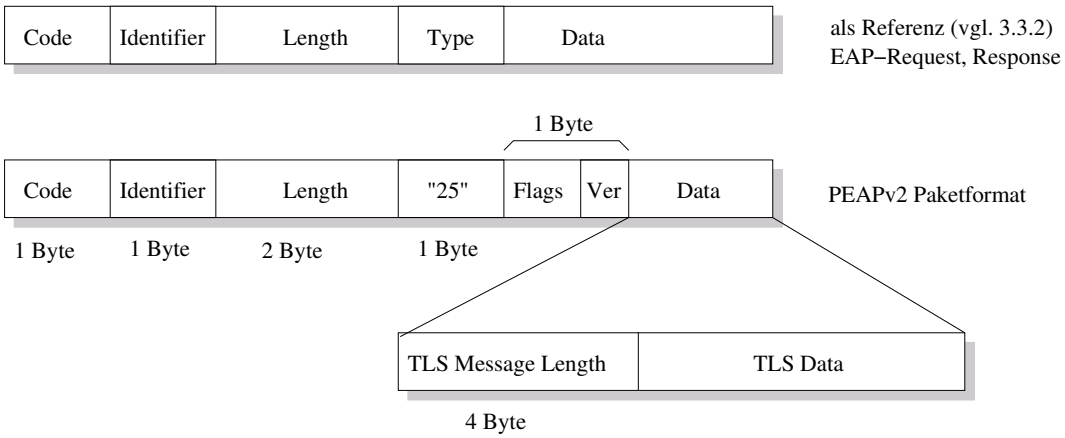


Abbildung 4.17: PEAPv2 Nachrichtenformat

Abbildung 4.17 zeigt das Nachrichtenformat von PEAPv2, wie es in Phase 1 verwendet wird. In der EAP-Request und Response Nachricht ist der Type für PEAPv2 mit 25 angegeben. Das erste Byte des Data Felds ist für 5 Bit Flags und 3 Bit Version reserviert. Das Flagfeld LMSSR setzt sich zusammen aus: Length, More Fragments, PEAP Start und zwei reservierten Bit.

Die Version ist "x10", also der Binärwert für Version 2. Wenn der Flag Length gesetzt ist, beinhalten Byte 2-5 des Data Feldes die Länge der mitgeführten TLS Records. Falls fragmentiert wird, dann ist die Länge über alle Fragmente anzugeben. In TLS Data werden nun TLS Records gekapselt.

Die Nachrichten der Phase 2 werden als Payload der EAP Methode EAP-TLV (s. Abbildung 4.18) transportiert. Dabei wird zwischen EAP Paketen der inneren Methode(n) und sonstigen Nachrichten kein Unterschied gemacht. Im Aufbau an das Paketformat aus Phase 1 angelehnt beinhaltet das TLS Datenfeld verschlüsselte EAP-TLVs. EAP-TLVs sind selbstbeschreibend, so dass bei PEAPv2 der EAP Header des EAP-TLV Pakets ausbleiben kann. Das heisst also, der Payload eines EAP-TLV ist eine Liste von TLVs ("TLV Tupel"). Die Spezifikation von PEAPv2 illustriert den Aufbau ([7], 3.4) durch:

```
PEAP Part 2 packet format = EAP-TLV [EAP-Payload TLV [[EAP packet],
TLVs, ...], TLVs, ...] OR TLS Alert
```

Demzufolge ist jede Nachricht in Phase 2 eine EAP-TLV Nachricht, die im Payload

- eine EAP Nachricht, möglicherweise gefolgt von TLVs, in einem EAP-Payload gekapselt
- und/oder TLVs

beinhaltet. Wichtig ist der Hinweis, dass nicht zwangsläufig eine EAP Paket gekapselt sein muss. In einigen Nachrichten kommen nur TLVs vor (vgl. Abbildung 4.15). Der Draft verweist explizit auf diesen Umstand, "The EAP-Payload is an (optional) TLV which encapsulates EAP packets (including all EAP header fields) and in addition carries TLVs associated with them."

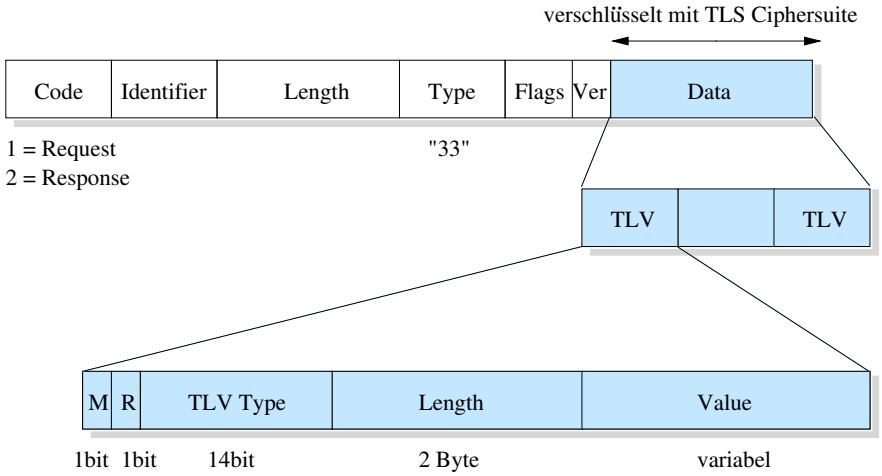


Abbildung 4.18: EAP-TLV Nachrichtenformat (Request, Response)

Code	Typ	verpflichtend
0-2	Reserviert	-
3	RESULT TLV - Acknowledged Result	ja
4	NAK TLV	ja
5	Crypto-Binding TLV	ja
6	Connection-Binding TLV	nein
7	Vendor-Specific TLV	ja (!)
8	URI TLV	nein
9	EAP Payload TLV	ja
10	Intermediate Result TLV	ja

Tabelle 4.2: PEAPv2 TLV Typen

Die Wahl von TLV ermöglicht, um obige Ausführungen zusammenzufassen, beliebige Informationen zwischen Client und Authentication Server zu transportieren, ganze EAP Nachrichten, Benachrichtigungen oder kryptographisches Material. Damit ist Herstellern im Nachhinein Gelegenheit gegeben, eigene TLVs ihrer PEAP Implementierung hinzuzufügen. Genau diesen Vorteil bietet ja auch RADIUS mit dem Konzept der Attribute.

Tabelle 4.2 listet die zur Verfügung stehenden TLV-Tupel Formate auf. Ist beispielsweise durch Code 9 ein EAP-Payload TLV signalisiert, wird im Value Feld das gesamte EAP

Paket, also einschließlich seiner Header (Code, Identifier, Length, Type) gekapselt. Im Anschluss daran können dem Value Feld weitere TLV Tupel folgen.

4.7.5 Schlüsselableitung

Bei der Beschreibung des Man-in-the-Middle Angriffs in Abschnitt 4.8 wurde lediglich der Ansatz des Auswegs genannt, nämlich das Schlüsselmaterial der inneren Methode mit dem der äußeren, also dem TLS Master Secret, kryptographisch zu verbinden. Diesem Problem ist das Internet-Draft "The Compound Authentication Binding Problem" ([1]) gewidmet und zeigt eine Lösung auf, die PEAPv2 auch übernommen hat.

Es wird also aus dem Schlüsselmaterial aller im Authentifizierungsprozess involvierten EAP Methoden, d.h. der für den TLS Tunnel verantwortlichen äußeren Methode wie allen im Tunnel ausgeführten Methoden, ein gemeinsamer Schlüssel generiert, dessen Besitz die jeweilige Partei als die "echte" ausweist.

Ziel ist, dass Client und EAP Server zwei Schlüssel ableiten:

Compound MAC Dieser Hash wird von den Schlüsseln aller inneren Methoden und den Tunnel Keys abgeleitet. Er ist gegen Man-in-the-Middle Angriffe geschützt, da der Angreifer nicht in den Besitz der Schlüssel der inneren Methoden gelangt, und kann dadurch zur Validierung der Identität dienen.

Compound Session Key Dieser Schlüssel dient der Erfüllung der Anforderung an eine EAP Methode, bestimmtes Schlüsselmaterial für den Export zu generieren.

Wenn alle Methoden im Tunnel abgeschlossen sind, erfolgt ein 2-Wege Handshake zwischen Client und Authentication Server (durch letzteren initiiert), in welchem die jeweiligen Compound MACs ausgetauscht werden (vgl. auch Abbildung 4.15):

Client <— Binding Request (B1) [S_NONCE ... B1_MAC]
Binding Response (B2) [C_NONCE ... B2_MAC] —> Authentication Server

Der Server sendet einen Binding Request B1 mit einem B1_MAC, welchen der Client validiert. Ist die Überprüfung erfolgreich, wird mit der Binding Response B2 ein B2_MAC im Payload zurückgeschickt. Kann auch der Server die Überprüfung erfolgreich durchführen, sind beide Parteien in der Gewissheit, keinen MitM als Gesprächspartner zu haben.
S_NONCE und C_NONCE sind zwei 32 Byte Zufallszahlen. Erstere geht in die Berechnung beider MACs ein, B1_MAC und B2_MAC, letztere nur in die Berechnung von B2_MAC.

Nun soll die Herleitung des Compound MAC gezeigt werden. Jede EAP Methode hat per Definition drei mindestens 64 Byte lange Schlüssel zu exportieren, MSK, EMSK und IV, wobei hier nur der MSK von Interesse ist. In der Spezifikation von PEAP, die sich an die Vorgaben von [1] hält, wird der jeweilige MSK durch TK bzw. durch ISK bezeichet.

Bezeichne also TK den MSK, der in Phase 1 aus dem TLS Master Secret abgeleitet wird. Es sind (vgl. 4.5.2) die ersten 64 Byte des 128 Byte langen Hashwerts

PRF (master secret, "client EAP encryption", Client.Random | Server.Random).

In jeder erfolgreichen inneren EAP Methode wird ebenfalls ein 64 Byte langer MSK generiert. Für die Methoden 1 bis n werde dieser durch $\text{ISK}_1, \ldots, \text{ISK}_n$ (Inner Session Key) bezeichnet, wobei ISK_i="" gesetzt wird, falls Methode i kein Schlüsselmaterial erzeugt. Das ist zum Beispiel bei EAP-MD5 der Fall.

Nach jeder erfolgreich beendeten inneren EAP Methode k wird der 32 Byte lange Intermediate Combined Key berechnet, wobei von der in TLS v1.0 definierten P-SHA1 Hashfunktion Gebrauch gemacht wird (vgl. hierzu Fußnote 8).

$\text{IPMK}_0 = \text{TK}_{(0...31)}$
for j=1 to k do
  IPMK_j = P-SHA1 (IPMK_{j-1}, "Intermediate PEAP MAC key" | ISK_j)

Dieser Algorithmus zeigt deutlich, dass sich die Schlüssel der inneren EAP Methoden untereinander verzahnen sowie eine Verbindung zum Schlüsselmaterial des TLS Tunnels aufweisen. Ist die letzte innere EAP Methode erfolgreich ausgeführt, werden aus IPMK_n zwei Schlüssel abgeleitet:

Die Berechnung des Compound MAC Key für den Server (B1_MAC) bzw. für den Client (B2_MAC), beide in der Länge von 20 Byte, erfolgt durch

CMK_{B1} = P-SHA1 (IPMK_n, "PEAP Server B1 MAC key" | S_NONCE)

CMK_{B2} = P-SHA1 (IPMK_n, "PEAP Client B1 MAC key" | C_NONCE | S_NONCE)

Der Compound Session Key (CSK) wird auf beiden Seiten erst abgeleitet, nachdem die abgesicherte Terminierung erfolgreich abgeschlossen wurde:

CSK = P-SHA1 (IPMK_n, "PEAP compound session key" | C_NONCE | S_NONCE | OutputLength)

Und an dieser Stelle schließt sich der Kreis: Der CSK hat eine Länge von mindestens 128 Byte zu haben, wovon die ersten 64 Byte der MSK, die zweiten 64 Byte der EMSK sind. Diese beiden Schlüssel werden zum AAA-Key transformiert (vgl. Abbildung 3.15 auf Seite 29), der auf Client und Authenticator als Ausgangspunkt für die Generierung kurzlebiger Sitzungsschlüssel herangezogen wird.

4.7.6 Sicherheitsbetrachtungen

Wie in jedem Internet Draft einer EAP Methode gibt es auch bei PEAPv2 ein Kapitel "Sicherheitsbetrachtungen" (security considerations), in dem Überlegungen zur Sicherheit der Methode angestellt werden. Insbesondere wird überprüft, ob die Anforderungen für eine im WLAN-Umfeld eingesetzte EAP Methode erfüllt werden. Daraus sollen drei Punkte aufgegriffen werden.

NaK Spoofing Nach dem anfänglichen Identity-Nachrichtenaustausch in Phase 1 unterbreitet der Authentication Server dem Client den Vorschlag, als Authentifizierungsmethode PEAPv2 zu verwenden. Für den Fall, dass der Client ein Verfahren nicht unterstützt, ist die EAP-Nak Nachricht vorgesehen. Bevor nicht der TLS Tunnel eingerichtet ist, werden alle Nachrichten im Klartext, d.h. ohne Sicherung der Authen-

tizität und Schutz der Integrität, verschickt. EAP-Nak gehört dazu. Der Angreifer kann also ein Zustandekommen einer PEAPv2 Authentifizierung durch anhaltendes Einstreuen von Nak-Nachrichten erschweren[6].

TLS Session Cache Bei der Implementierung der Sitzungswiederaufnahme (session resuming) muss der Cache der TLS Sitzungen besonders beachtet werden. In PEAPv2 wird der Ausgang einer Authentifizierung, wie oben bereits genannt, nicht durch Verschicken einer EAP-Success bzw. Failure Nachricht dem Client bekanntgegeben. Eine Beendigung des Authentifizierungsvorgangs erfolgt erst nach gegenseitiger Verständigung auf Beendigung der Sitzung, durch den "protected termination" Mechanismus. Verläuft nun Phase 1, die Einrichtung des TLS Tunnels, erfolgreich, Phase 2 schlägt aus welchen Gründen auch immer fehl, so muss die Implementierung dafür sorgen, dass der TLS Sitzungscache - wenn nicht (korrekterweise) bereits geleert - nicht dafür herangezogen wird zu ermitteln, ob der Client bereits authentifiziert wurde.

EAP-Success, EAP-Failure Tatsache ist, dass diese Nachrichten nicht kryptographisch geschützt sind und somit von einem Angreifer ohne Gefahr, dabei erkannt zu werden, eingestreut werden können. Bei PEAPv2 wird diesen Nachrichten aufgrunddessen keine große Bedeutung mehr beigemessen. Innerhalb des TLS Tunnels gibt der Server dem Client das Ergebnis der Authentifizierung bekannt, welcher mit einer Bestätigung antwortet. Diese Nachrichten werden verschlüsselt übertragen und bieten Schutz der Authentizität und Datenintegrität. Erst im Anschluss an diese gesicherte Beendigung darf der Client empfangene EAP-Success bzw. EAP-Failure Nachrichten bearbeiten, und das auch nur, wenn die EAP Nachricht mit der Entscheidung aus dem Tunnel übereinstimmt. In allen anderen Fällen sind die EAP-Success bzw. EAP-Failure Nachrichten stillschweigend zu verwerfen.

Zieht man nun ein Resümee und beurteilt PEAPv2 hinsichtlich seiner Verwendung in WLAN, so fällt dies durchaus positiv aus. Einen relativ großer Teil der Spezifikation entfällt auf Sicherheitsbetrachtungen, welche auch auf das Bedrohungsmodell aus Abschnitt 4.1 eingeht. Es scheint mit PEAPv2 eine gut durchdachte EAP Methode vorzuliegen, die ein Maximum aus der Kombination von Sicherheit (TLS-Tunnel), Einfachheit (passwortbasierte innere Methode z.B.) und Flexibilität (TLV) erreichen will. Vom technischen Standpunkt, der in diesem Rahmen natürlich nur gestreift werden konnte, zeigt sich PEAPv2 gut gerüstet für zukünftige Anwendungen. Es ist abzuwarten, ob Cisco seine Zusammenarbeit mit Microsoft an PEAPv2 beenden und eigene Authentifizierungsverfahren verstärkt fördern wird und inwiefern sich dieser Umstand auf PEAPv2 auswirkt.

4.8 EAP-TTLS

EAP-TTLS (Tunneled TLS) ist 2001 von Funk Software und Certicom auf den Markt gekommen und befindet sich wie die meisten anderen EAP Methoden noch in der Entwicklungsphase. Da es relativ viele Gemeinsamkeiten mit PEAP aufweist, wird einer detaillierten Protokollbeschreibung ein Herausheben der Besonderheiten vorgezogen. EAP-TTLS wird von vielen Implementierungen unterstützt und ist relativ weit verbreitet.

[6] Bei solchen Denial-of-Service Angriffen ist prinzipiell zu beachten, dass der Zeitraum, in dem sie zu einem möglichen Erfolg führen, äußerst gering ist, kein Zeitintervall von Sekunden. Als entferntes Beispiel sei die von der ETSI vorgegebene maximale Gesamtzeit einer GSM Authentifizierung genannt, sie hat in weniger als 500 ms (!) zu erfolgen.

Die innere Authentifizierung kann zum einen durch beliebige EAP Methoden, zum anderen aber auch durch herkömmliche passwortbasierte Verfahren wie PAP, CHAP, MS-CHAP oder MS-CHAP-V2 erfolgen. EAP-TTLS bietet diese Flexibilität als einzige EAP Methode an. Unmittelbar daraus ergibt sich ein großer Vorteil, dass nämlich bestehende Benutzerdatenbanken beibehalten werden können, was unter anderem Internetanbietern (ISPs) zugute kommt. Da die Authentifizierung durch den infolge des TLS Handshakes etablierten sicheren Tunnel geleitet wird, sind diese an sich unsicheren Verfahren kryptographisch hinreichend gesichert. Wörterbuch- und Man-in-the-Middle Angriffe werden zumindest stark erschwert.
Die derzeitig aktuelle Draft (Version 3) vom Stand August 2003 geht ausführlich auf die Gefahren ein, die durch Anwendung von EAP-TTLS im drahtlosen Umfeld hinzukommen und bietet zu den einzelnen Punkten Gegenmaßnahmen und Sicherungsmechanismen an.

Wie auch bei PEAPv2 wird die Authentifizierung in zwei Phasen unterteilt. In Phase 1 wird ein TLS Handshake ausgeführt, in dem serverseitige Authentifizierung, Generierung gemeinsamen Schlüsselmaterials und Verhandlung einer TLS Ciphersuite stattfindet, durch welche die nachfolgende Kommunikation geschützt wird. Optional kann in Phase 1 auch die clientseitige Authentifizierung durchgeführt werden, was allerdings ein Zertifikat samt privatem Schlüssel voraussetzt. Somit bietet EAP-TTLS - wie PEAP - auch einen "EAP-TLS" Betriebsmodus. Sollte diese Option in Anspruch genommen worden sein, kann die durch die verhandelte TLS Ciphersuite geschützte Phase 2 weitere Authentifizierungen beinhalten. In dieser Hinsicht wird das von EAP-TTLS verfolgte Designziel größtmöglicher Flexibilität besonders deutlich.

Vergleicht man den Umfang der Spezifikationen von PEAPv2 und EAP-TTLS (72 zu 41 Seiten) so lässt dies die Vermutung auf fehlende Sicherungsmechanismen zu. Dies ist in der Tat der Fall, da zum Beispiel weder der "protected termination" Mechanismus am Ende von Phase 2 noch eine kryptographische Bindung des Schlüsselmaterials berücksichtigt wird. Also ist zumindest die Gefahr des Einspielens von ungeschützten EAP-Success bzw. Failure Nachrichten, die einem Denial-of-Service Angriff dienen, nicht unterbunden. Ob nun der in [As01] präsentierte Man-In-The-Middle Angriff bei EAP-TTLS realistisch ist oder andere Mechanismen davor schützen, würde in eine Protokollanalyse münden.

Als eine Besonderheit der Spezifikation fällt die durchgehende Berufung auf vier am Authentifizierungsprozess beteiligten Parteien auf. Der (Authentication) Server ist hierbei in zwei Einheiten geteilt, und zwar in den TTLS Server und den AAA/H Server, die auch in den Beispiel-Nachrichtenabläufen übernommen sind. Damit ist der Server, der EAP-TTLS implementiert hat und mit dem Client die Authentifizierungsverhandlung durchführt, von dem Backendserver, der die Authentifizierung und Authorisierung des Clients steuert, getrennt. Natürlich ist diese Trennung nicht zwangsläufig auch physisch, so dass der TTLS Server durchaus auch die Benutzerauthentifizierung durchführen kann. Im anderen Fall agiert er als Proxy zwischen Authenticator und AAA/H Server.

In TTLS Phase 2 werden Nachrichten im AVP-Format (attribute-value pairs) transportiert, das über die die Authentifizierung betreffenden Nachrichten hinaus einen Austausch beliebiger Informationen zwischen Client und TTLS Server ermöglicht. Als Beispiele für Informationen werden Sicherheitsparameter, Schlüsselmaterial und Accounting Informationen angegeben, wobei sich beliebige weitere Attribute hinzufügen lassen. Die damit erzielte Flexibilität wird von PEAPv2 durch das TLV (Type-Length-Value) Nachrichtenformat in Phase 2 ebenfalls erreicht.

Nach Firmenangaben wird in Kürze eine überarbeitete Spezifikation von EAP-TTLS mit erheblichen Erweiterungen veröffentlicht. Es ist zu erwarten, dass sie allen bekannten Gefährdungen einer EAP Methode hinreichende Mechanismen entgegenstellt, so dass eine Bewertung an dieser Stelle noch nicht zu treffen ist.

4.9 EAP-FAST

EAP-FAST (Fast Authentication via Secure Tunneling) ist eine sehr neue EAP Methode. Sie ist von Cisco Systems entwickelt und Anfang Februar 2004 bei der IETF veröffentlicht worden. Wie PEAPv2 und EAP-TTLS ist auch EAP-FAST eine getunneltes Verfahren, verfolgt also das Ziel, die Einfachheit von Benutzername/Passwort-basierter Authentifizierung mit der Sicherheit eines TLS Tunnels zu vereinen. In der Tat scheint mit diesem Verfahren ein Schritt in die richtige Richtung getan, da die Beliebtheit passwortbasierter Authentifizierungsverfahren nach wie vor ungebremst ist. Weiterhin fallen unter die Designziele die Minimierung des erforderlichen Konfigurationsaufwands wie des Rechenaufwands durch Verwendung effizienter kryptographischer Verfahren. Damit richtet sich EAP-FAST insbesondere an Anwendungen im Wireless Umfeld. Diese fordern in der Regel einen unkomplizierten Ad-Hoc Zugang zum Netzwerk. Weiterhin weisen die in diesem Bereich eingesetzten Geräte meistens eine geringe Rechenleistung und beschränkte Energiereserven auf.

EAP-FAST ist nicht zuletzt als Ergänzung des eigenen Produktportfolios gedacht und dem Schaffen einer Alternative zu dem mit Schwächen behafteten LEAP. Da im Gegensatz zu LEAP die Spezifikation aber veröffentlicht wurde, erhofft sich Cisco natürlich eine Verbreitung über die Firmengrenzen hinaus. Eine Nachfrage in der Patentabteilung von Cisco Systems hat bestätigt, dass EAP-FAST frei verfügbar und ohne Entrichtung jeglicher Lizenzgebühren in eigenen Implementierungen zu vertreiben ist.

Auf den ersten Blick sind wesentliche Teile der Spezifikation von PEAPv2 übernommen worden, so dass das Herausfinden der Neuerungen an vorderster Stelle steht. Berechtigt ist auch die Frage, warum nicht alle Anstrengungen auf die Weiterentwicklung von PEAPv2, der Gemeinschaftsarbeit mit Microsoft, gerichtet wurden. In [Ci04a] wird hierauf indirekt eine Antwort gegeben, indem als Termin, ab dem EAP-FAST in Produkten zur Verfügung steht, das erste Quartal 2004 genannt wird. Offensichtlich ist schnellstmöglichst eine akzeptable, mehr Sicherheit bietende Lösung benötigt worden, was möglicherweise bei PEAPv2 nicht in der Kürze zu erreichen war.

4.9.1 Neuerungen

In einem vorgezogenen Schritt, der im folgenden mit Phase 0 oder Provisioning Phase bezeichnet wird, wird der Client mit einem geheimen, symmetrischen Schlüssel ausgestattet, einem nur von Client und EAP Server geteilten Wissen. Dieser ist im PAC (Protected Access Control) integriert. Der Besitz eines PAC ist Voraussetzung für einen überhaupt stattfindenden Authentifizierungsvorgang und setzt sich aus drei Elementen zusammen:

- PAC Key: Zwischen EAP-Server und Client geteilter symmetrischer Schlüssel (32 Byte Zufallswert), der für die Tunneleinrichtung in Phase 1 sowie die gegenseitige Authentifizierung herangezogen wird. Er wird an entsprechender Stelle auf das TLS Premaster Secret abgebildet, womit die zu dessen Ableitung eigentlich nötigen Berechnungen ausbleiben können.

- PAC Opaque (variable Länge): PAC Opaque (engl. undurchsichtig, verschleiert) sind vom PAC Server generierte, im Aufbau und Inhalt nur ihm bekannte Daten. Der Client hat diese interpretationslos abzuspeichern und auf Auforderung hin zu schicken. Der Opaque Anteil dient dem Server zur Überprüfung, ob der Client im Besitz des PAC Key ist.
- PAC Info (variable Länge): Beinhaltet Angaben zum Herausgeber des PAC oder eine Authority Identity (A-ID), oder auch die Lebensdauer des PAC.

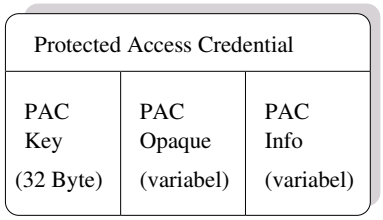


Abbildung 4.19: Protected Access Credential (PAC)

Bereits am Tage der Veröffentlichung von EAP-FAST ist die Provisioning Phase in die Kritik geraten, weil sie einen erfolgreichen Man-in-the-Middle Angriff in Aussicht stellt. Einer Untersuchung der PAC Zustellung in Phase 0 hat also besondere Beachtung geschenkt zu werden. In diesem Zuge soll auch der Angriff im Detail und seine Aussichten auf Erfolg vorgestellt werden. Bereits an dieser Stelle muss darauf hingewiesen werden, dass der PAC auch im Vorwege verteilt sein oder durch anderweitige Verfahren auf den Rechner des Clients transferiert werden kann - die Kritik richtet sich also nur gegen eine der vielen Varianten. Die Spezifikation von EAP-FAST geht ausführlich auf diese Variante und ihre mögliche Gefährdung ein.

Neben Erfüllung der Anforderungskriterien für den Gebrauch in Wireless LANs (vgl. [3]) werden insbesondere die Flexibilität in Folge der Unterstützung zahlreicher gängiger Passwortverfahren wie OTP, MS-CHAPv2 und LDAP, sowie die Effizienz des Verfahrens, im Sinne von Rechenaufwand, angeführt.

Weiterhin im Vordergrund steht ein neues Konzept, das Verteilen eines Premaster Secrets (PAC; s.o.) infolge rechenintensiver asymmetrischer Verfahren und das Ableiten des Master Secrets mit schnellen symmetrischen Verfahren. Dabei greift Cisco auf aktuelle Forschungsergebnisse aus dem universitären Umfeld zurück ([16]). In dieser Quelle findet sich auch die Angabe, dass auf einem mobilen, rechenschwachen Gerät ein vollständiger TLS Handshake mit einem 1024bit RSA Schlüssel bis zu 30 Sekunden in Anspruch nimmt.

In der Konzeption besteht EAP-FAST aus zwei Protokollen mit vollkommen unterschiedlichen Absichten:

- Bereitstellungsprotokoll (Provisioning): Durch einen eingerichteten TLS Tunnel lässt der Server dem Client das starke Credential, den PAC, zukommen. Diese Zustellung erfolgt in der Regel pro Client nur einmal, da EAP-FAST für nachträgliche Änderungen eine Update-Funktion im Protokoll vorgesehen hat. In der Provisioning Phase ist schwergewichtige, asymmetrische Kryptographie vorherrschend.
- (Benutzer-)Authentifizierung. Sie erfolgt jedesmal, wenn der Client Zugang zum Netzwerk erlangen möchte. Dabei wird zur Steigerung der Effizienz symmetrische Kryptographie herangezogen.

4.9.2 Protokollbeschreibung

Die Authentifizierung wird wie bei EAP-TTLS oder PEAP auch in zwei Phasen unterteilt:

- Phase 1: Einrichtung des Tunnels
- Phase 2: Getunnelte Authentifizierung

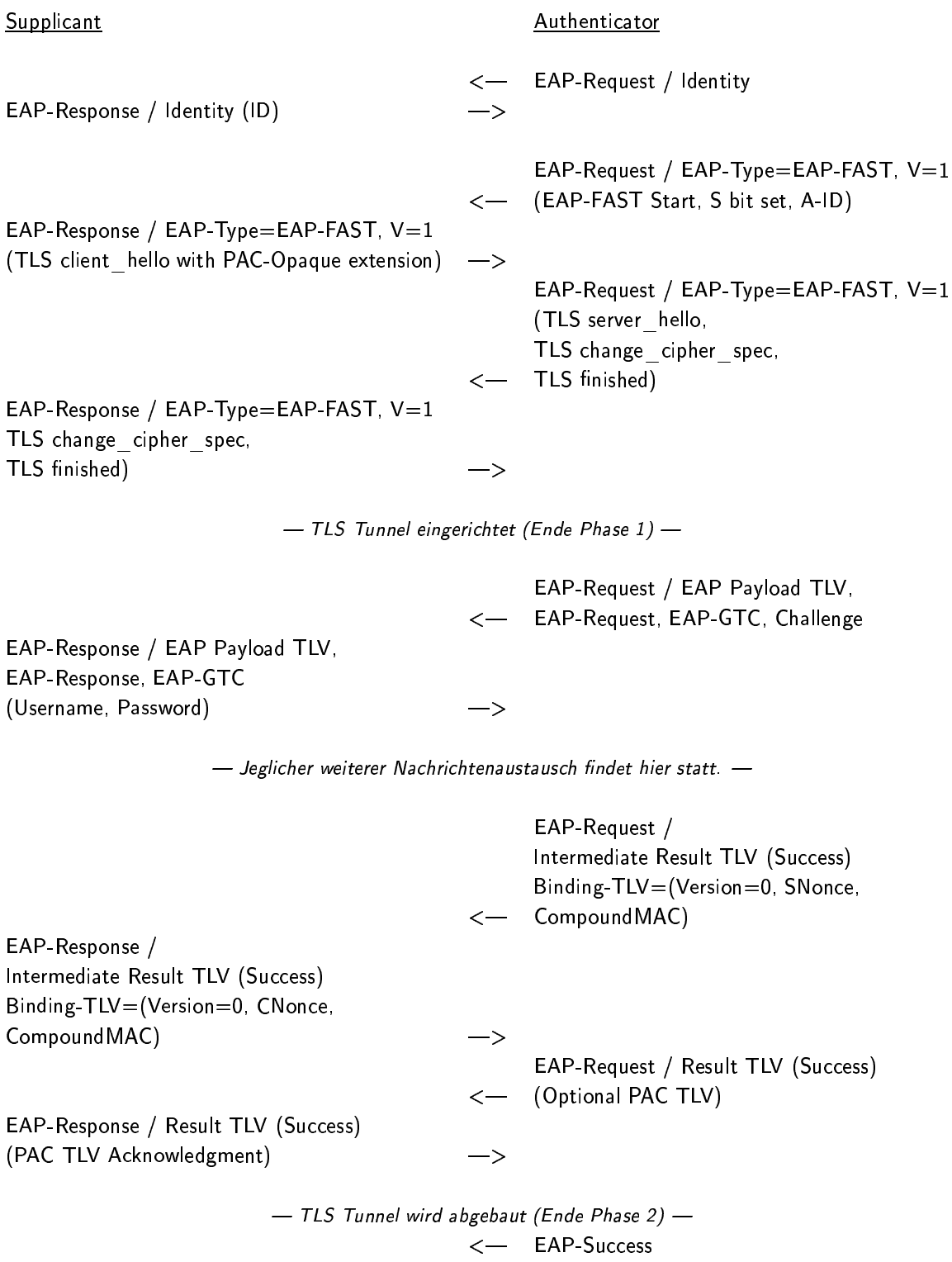


Abbildung 4.20: EAP-FAST Authentifizierung (Phase 1 und 2)

Zur Phase 1:

Die aus dem Konzept des PAC resultierenden Vorteile werden in Phase 1 besonders deutlich. Der TLS Handshake, an dessen Ende ein sicherer Tunnel steht, kommt mit wesentlich weniger Nachrichten in Form von TLS Records aus. Das vom Server geschickte EAP-FAST/Start Paket beinhaltet dessen A-ID (Authority Identity). Der Client sucht aus der lokalen PAC Datenbank den korrespondierenden PAC und transferiert in einer Erweiterung des TLS Client_Hello den PAC-Opaque Anteil zum Server. Existiert beim Client noch kein zu dieser A-ID gehörender PAC, so hat der Client durch Schicken einer "konventionellen" Client_Hello Nachricht die Provisioning Phase einzuleiten, um in den Besitz des erforderlichen PAC zu kommen.

In der ersten Version von EAP-FAST ist die Anzahl der für Phase 1 zur Verfügung stehenden TLS-Ciphersuites auf eine beschränkt, und zwar `TLS_RSA_WITH_RC4_128_SHA`. Das Interesse gilt dem 128bit RC4 und SHA1 - der RSA Schlüsselaustausch wird nicht in Anspruch genommen.

Bezüglich des am Ende von Phase 1 vorliegenden Schlüsselmaterials ist ein 40 Byte Session Key Seed SKS hinzugekommen, der in Phase 2 benötigt wird. Das Master Secret wird abgeleitet aus Client.Random, Server.Random und dem PAC-Key unter Anwendung der EAP-FAST eigenen Hashfunktion T-PRF, die wie TLS-PRF einen Hash in beliebiger Länge generieren kann.

Ihre Ausgabe wird in diesem Fall auf 48 Byte Länge gesetzt. Als Premaster Secret wird, wie zuvor bereits erwähnt, der von beiden Seiten geteilte PAC Key verwendet.

Master Secret	=	T-PRF(PAC-Key, "PAC to master secret label hash", Server.Random | Client.Random, 48)
Key Block	=	PRF(Master Secret, "key expansion", Server.Random | Client.Random)

EAP-FAST generiert nun aus Key Block das im EAP Schlüssel-Framework [2] vorgegebene Schlüsselmaterial, und zwar den MSK, EMSK und IV sowie den Session Key Seed.

Zur Phase 2:

EAP-FAST Phase 2 ist sehr ähnlich der von PEAPv2. Ein Unterschied besteht in der Forderung, dass mindestens eine innere Methode auszuführen ist, während PEAPv2 auch die Möglichkeit bietet, ohne innere EAP-Methode auszukommen. Dies resultiert aus dem Angebot von PEAPv2, eine gegenseitige Authentifizierung bereits in Phase 1 stattfinden zu lassen, sofern nämlich der Client im Besitz eines Zertifikats und dieses auch zu schicken bereit ist.

Auch EAP-FAST verwendet für den Transport von Nachrichten innerhalb des Tunnels ein generisches TLV Format, zum Beispiel werden EAP Nachrichten in EAP Message TLVs gekapselt. Sollte ein Update des PAC erfolgen, so kann dies in Phase 2 geschehen, wofür ein eigener PAC TLV vorgesehen ist. Das Ende der Phase 2 wird durch eine abgesicherte Beendigung wie bei PEAPv2 besiegelt.

Auch EAP-FAST leistet der Aufforderung nach kryptographischer Bindung des Schlüsselmaterials Folge, indem in Phase 2 nach jeder erfolgten inneren Methode ein Compound

Key bestimmt wird. Hierzu wird der aus Phase 1 exportierte 40 Byte Session Key Seed herangezogen. Von dem erzeugten Schlüsselmaterial der jeweiligen inneren EAP Methode werden die ersten 32 Byte des Session Keys herangezogen, die im folgenden mit $\text{ISK}_1, \ldots, \text{ISK}_n$ bezeichnet werden. Sollte eine innere Methode i kein Schlüsselmaterial generieren, so sind die 32 Byte mit Nullen (0x00) zu füllen. Der Compound (Session) Key wird durch

$\text{S-IMCK}_0 = \text{SKS}$
for $j = 1$ to $n - 1$ do

$\text{IMCK}_j = \text{T-PRF}$ (S-IMCK_{j-1}, "Inner Methods Compound Keys", ISK_j, 60)

generiert, wobei S-IMCK_j die ersten 40 Byte von IMCK_j repräsentiert. Von den von T-PRF generierten 60 Byte Ausgabehash gehen jeweils die ersten 40 Byte als Eingabe in die nachfolgende ICMK_{j+1} Ableitung ein. Die verbleibenden 20 Byte werden als Schlüssel CMK_j verwendet. Dieser dient der Generierung des Intermediate Crypto-Binding Compound MAC.

Das kryptographische Aufeinanderbeziehen der Sitzungsschlüssel wird unmittelbar deutlich. Diese Berechnung führen Client und Server eigenständig und unabhängig voneinander durch. Damit ist sichergestellt, dass ein Außenstehender sich nicht in die Situation bringen kann, den Nachrichtenaustausch von Phase 2 zu entschlüsseln.

Die Berechnung ist in der Spezifikation absichtlich allgemein gehalten, durch Einführung der Variable n, die die Anzahl der im TLS Tunnel durchgeführten inneren EAP Methoden steht. Im praktischen Einsatz wird es sicherlich der Regelfall sein, eine einzige[7] Authentifizierung im Tunnel durchzuführen. .

In Phase 1 erfolgt üblicherweise die Authentifizierung des Servers. Durch den dann eingerichteten TLS Tunnel hat demzufolge nur noch die Authentizität des Clients überprüft zu werden, beispielsweise durch einmalige Anwendung von EAP-MSCHAPv2, worin der Client seinen Benutzernamen nennt und den Besitz des Passworts beweist.

Ist Phase 2 nach Erfolgen der abgesicherten Beendigung abgeschlossen, muss das zu exportierende Schlüsselmaterial, der Master Session Key, generiert werden. Dieser MSK geht in der geforderten Länge von 64 Byte aus dem letzten Compound Key, genauer: dessen ersten 40 Byte, hervor:

$\text{MSK} = \text{T-PRF}$ (S-IMCK_n, "Session Key Generating Function", 64)

Somit ist er ein Ergebnis aller EAP Methoden, die einen IMCK generieren.

4.9.3 Phase "0" - EAP-FAST provisioning

Es ist zu klären, wie der Client in den Besitz des PAC kommt, der Voraussetzung für eine überhaupt stattfindende Authentifizierung mit EAP-FAST ist. Es liegt am Client, diese Phase einzuleiten, und zwar genau dann, wenn zu dem empfangenen A-ID kein entsprechender PAC gefunden werden kann. Der Server erkennt das Einleiten der

[7] Durch Setzen von $n = 1$ macht die Formel zur Berechnung von IMCK_j aber keinen Sinn. Wahrscheinlich ist `for j=1 to n` gemeint.

Provisioning Phase, indem die TLS Client_Hello Nachricht keinen PAC Opaque mitführt.

In EAP-FAST ist ein Mechanismus für die Zustellung des PAC integriert, der aber nicht verpflichtend in Anspruch zu nehmen ist. Der Vorteil dieser "in-band" Variante ist, dass lediglich Benutzername und Passwort zur Echtheitsbestätigung der Identität erforderlich sind. Um größtmögliche Flexibilität hinsichtlich der Rechenbelastung des Clients zu erreichen, wird bewusst eine schwächere kryptographische Sicherung in Kauf genommen.

Wie in EAP-FAST Phase 1 wird ein TLS-gesicherter Tunnel aufgebaut, wofür zwei Ciphersuites zur Verwendung angeboten werden, zum einen

- `TLS_DH_anon_WITH_AES_128_CBC_SHA`, zum anderen
- `TLS_DHE_RSA_WITH_AES_128_CBC_SHA`.

Im Ergebnis ist zwischen Client und Server ein TLS Tunnel eingerichtet. Die gewählte Ciphersuite wirkt sich nur auf dessen genaues Zustandekommen aus.

Anonymer Diffie-Hellman bietet den großen Vorteil, gänzlich ohne Zertifikate auszukommen - und es gibt viele Anwendungsgebiete, in denen keine Public Key Infrastruktur vorhanden ist. Unmittelbar daraus ergibt sich aber auch der Nachteil, dass nämlich weder die eine noch die andere Seite die Authentizität des Tunnelpartners überprüfen kann. Dadurch, dass der Client keine Möglichkeit hat, die Echtheit des Servers zu überprüfen, ist ein Man-in-the-Middle Angriff möglich.

Die zweite Variante lässt den Server mit einem Zertifikat ausstatten, so dass der Tunnel zumindest serverauthentifiziert ist. Durch diesen Tunnel erfolgt gegenseitige Authentifizierung mit einem beliebigen passwortbasierten Verfahren. Die Wahl ist nicht mehr auf MS-CHAPv2 wie bei der anonymen Diffie-Hellman Variante beschränkt.

Kann die Authentifizierung erfolgreich beendet werden, wird dem Client der PAC zugestellt.

Natürlich sind sich die Entwickler von EAP-FAST der Gefahr der `TLS_DH_anon` Ciphersuite bewusst und haben entsprechende Schutzmechanismen zur Minimierung des Risikos eingebaut. Zudem ist dessen Verwendung nur als eine Option definiert. Die Zustellung des PAC erfolgt in der Regel auch nur ein einziges Mal. Sollte nachträglich der PAC neu verteilt oder sonstwie verändert werden, ist das integrierte PAC Refreshing Protokoll vorgesehen.

Die Provisioning Phase ist, wie bereits erwähnt, unmittelbar nach der Veröffentlichung von EAP-FAST in die Kritik geraten, wozu eine Sprecherin von Cisco, Linda Horiuchi, die Vorwürfe wie folgt kommentiert ([Br03]):

> "Anonymous DH is an option for provisioning the credential to the client machine, not for authenticating the user. If anonymous DH is used for credential provisioning, it is likely to be used once, during initial provisioning, not with every authentication. Further, a dictionary attack on anonymous DH would have to be an active attack, not an offline attack. An organization that is concerned about a vulnerability during initial credential provisioning should use a mechanism other than unauthenticated DH for initial credential provisioning. However, many organizations may consider the exposure window so small that unauthenticated DH is a prudent choice."

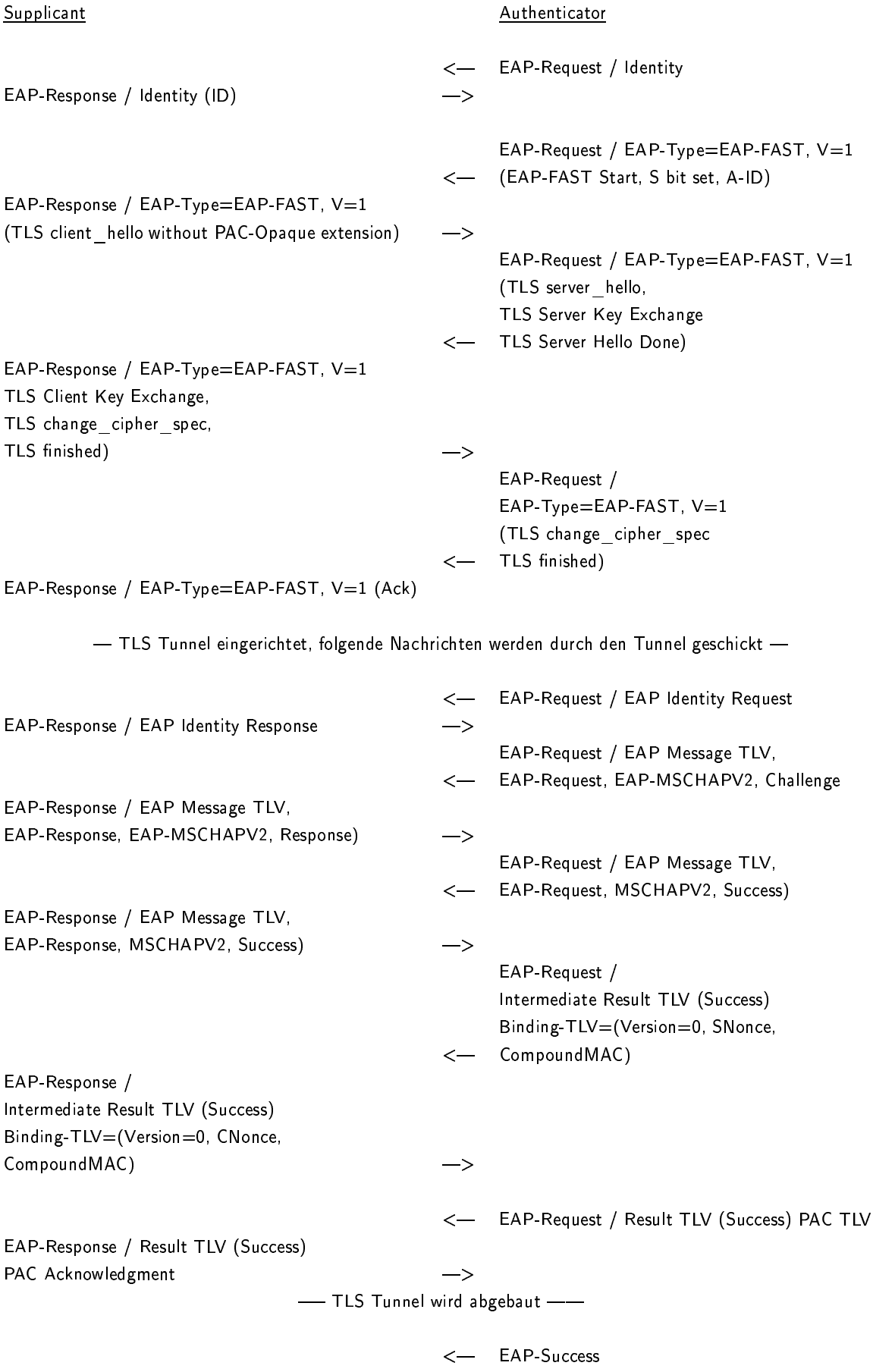


Abbildung 4.21: EAP-FAST Provisioning (Phase 0)

In der Tat ist der Zeitrahmen, in dem der Man-in-the-Middle Angriff Aussicht auf Erfolg hat, sehr gering. Es beginnt mit der Schwierigkeit, sich zwischen dem legitimierten Client und dem EAP Server zu positionieren. Ist dieser Angriff auch eher theoretischer Natur, soll die prinzipielle Vorgehensweise vorgeführt werden. Als Grundlage hierfür ist zunächst der Nachrichtenaustausch der Provisioning Phase zu studieren.

Verglichen mit der EAP-FAST Authentifizierung, d.h. den Phasen 1 und 2, werden mehr Nachrichten ausgetauscht, insbesondere aufgrund des vollständigen TLS Handshakes, der zum Aufbau des Tunnels nötig ist. Ist dieser eingerichtet, wird eine MS-CHAPv2 Authentifizierung hindurchgeleitet (4 Nachrichten), wonach eine gesicherte Beendigung und letzlich die Zustellung des PAC erfolgt.

Abbildung 4.22 zeigt das Einbringen des MitM in die Provisioning Phase. Die Abbildung wird zudem dazu genutzt, eine Vorstellung des ausgetauschten und generierten Schlüsselmaterials unter Verwendung der `TLS_DH_anon` Ciphersuite zu geben, wobei für den MitM lediglich der MS-CHAPv2 Challenge von Interesse ist. Da der Tunnelendpunkt nicht authentifiziert ist, hat der Client zunächst keine Möglichkeit, zwischen MitM und dem "echten" EAP Server zu unterscheiden. Dass die Kommunikation mit einem Angreifer stattfindet, wird frühestens bei der MS-CHAPv2 Authentifizierung im Tunnel aufgedeckt.

Neben dem TLS-üblichen Austausch von Nonces, Server.Random und Client.Random, haben Client und Server die entsprechenden Diffie Hellman Parameter zu generieren und der Gegenseite zuzustellen.

Der Server generiert einen Modulus p, eine primitive Wurzel g sowie einen privaten Schlüssel, X_S, wonach er das Tripel $(g, p, Y_S = g^{X_S} \mod p)$ an den Client sendet, welcher mit seinem privaten Schlüssel X_C nun $Y_C = g^{X_C} \mod p$ berechnet und dies zurückschickt. Beide kommen auf das gleiche TLS Premaster Secret, da

$$(Y_S)^{X_C} \mod p = (g^{X_S})^{X_C} \mod p = (g^{X_C})^{X_S} \mod p = (Y_C)^{X_S} \mod p$$

gilt. Aus dem Premaster Secret wird daraufhin das TLS Master Secret abgeleitet. Mit der in TLS definierten Pseudozufallszahlen-Funktion PRF wird zudem weiteres Schlüsselmaterial generiert, woraus 32 Byte (jeweils 16 Byte) für den Challengetext der MS-CHAPv2 Authentifizierung herangezogen werden. Hiermit soll die Zufälligkeit der beiden Challenges sichergestellt werden.

Der nun folgende MS-CHAPv2 Nachrichtenaustausch wird durch den Server (MitM) initiiert. Nach Empfang des Challenges berechnet der Client die Challenge Response, in die neben anderen Parametern das Passwort einfließt. Unter der Annahme, dass zu dem Challenge ein Eintrag in der vorberechneten Wörterbuchdatei des MitM vorhanden ist, ist dieser nun im Besitz des Passworts. Abhängig von der hierfür benötigten Zeit kann sogar dem Auslösen eines Timeouts zuvorgekommen und der vom Client geschickte Challenge sogar korrekt beantwortet werden.

Nach einem ersten Blick in die MS-CHAPv2 Spezifikation (RFC 2579) ist ein Gelingen allerdings nahezu auszuschließen. Die vom Server (MitM) zu berechnende 24 Byte Challenge Response (in MSCHAPv2: "NT-Response") benötigt nämlich als Eingaben den Benutzernamen, das Passwort, den Server Challenge sowie den Client Challenge. Letztere sind aber erst unmittelbar zuvor aus dem TLS Master Secret abgeleitet worden. Zudem ist ihre zufällige Zusammensetzung durch Anwendung von TLS-PRF garantiert.

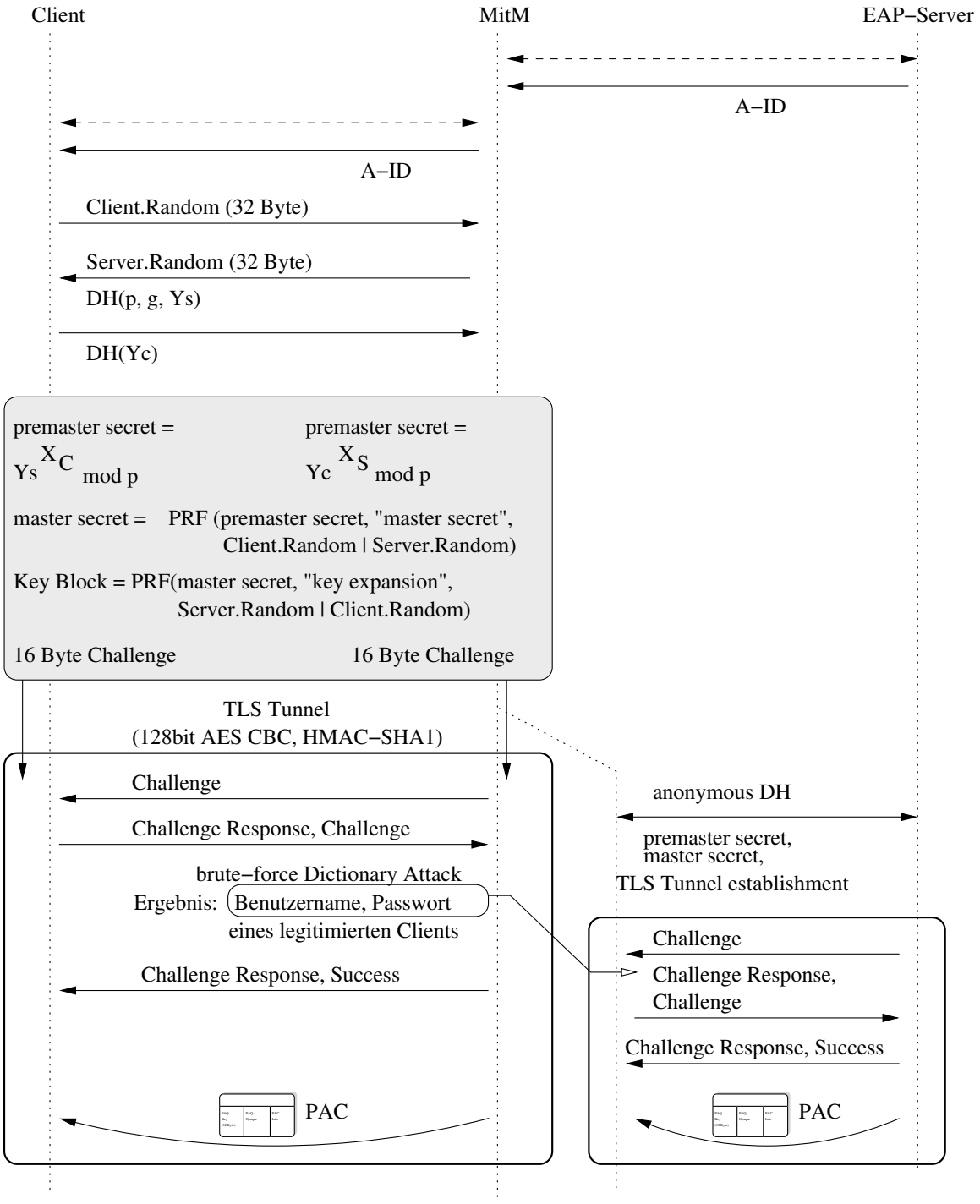


Abbildung 4.22: MitM in der Provisioning Phase

Festzustellen ist somit, dass ein Timeout auf Clientseite höchstwahrscheinlich einer gültigen Challenge Response zuvorkommt. Das Auslösen eines Timeouts ist somit von der Implementierung als relativ eindeutiger Hinweis auf Kompromittierung der Verhandlung zu deuten.

Angenommen aber, der MitM gelangt in den Besitz des Passworts. Durch die Kenntnis von Benutzername und Passwort eines legitimen Benutzers könnte er nun selbst die Provisioning Phase mit dem EAP Server initiieren (Abbildung 4.22). Am Ende der Verhandlung stünde dann die Zustellung des PAC.

Nun liegt es in der Hand der konkreten EAP-FAST Implementierung, wonach der MitM die Details seines Angriffs ausrichtet. Weiterhin spielt das Verhalten bzw. die Reaktion des Benutzers nach einem eingetretenen Timeout eine wichtige Rolle. Eine Möglichkeit für den MitM wäre zum Beispiel der Versuch, eine "Koexistenz"-Strategie zu verfolgen. Das heisst, dass die vom echten EAP Server empfangene A-ID an den Client weitergereicht wird, genauso wie der am Ende zugestellte PAC. Das hätte die Konsequenz, dass Client und MitM sich unter einer Identität ausgeben und - erneut der Verweis an die konkrete Implementierung - unabhängig voneinander Netzwerksitzungen aufbauen können.

Zusammenfassend ist aber der hohe theoretische Anteil an dieser Unternehmung festzustellen und die daraus resultierenden geringen Aussichten auf Erfolg. Durch die Wahl eines starken Passworts wird dessen Ableiten aus der Client Challenge Response noch unwahrscheinlicher. Dieser Forderung sollte generell bei passwortbasierten Authentifizierungsverfahren Beachtung geschenkt werden (vgl. LEAP).

Als Abschluss soll ein kurzer Blick auf den Session Resuming Fall geworfen werden, da dieser ebenfalls von dem Konzept des PAC profitiert.

4.9.4 Session Resuming

Im Falle von Session Resuming kann die gesamte Authentifizierungsphase 2 ausbleiben, also die Ausführung einer oder mehrerer innerer EAP Methoden. Der Server erkennt eine Anfrage auf Wiederaufnahme einer Sitzung anhand des Empfangs einer bekannten Session ID, die Bestandteil der vom Client geschickten Client_Hello Nachricht ist. In diesem Fall ist auch der PAC Opaque Anteil in der Client_Hello Erweiterung auszulassen.

Auf Seiten des Servers wird der Tupel (Client Session ID, Master Secret, CipherSpec) aus dem Cache geholt und die Entscheidung über Wiederaufnahme der Sitzung gefällt. Steht einer Wiederaufnahme nichts im Wege, wird dem Client dies durch Setzen dieser Session ID in der TLS Server_Hello Nachricht signalisiert. Anschließend wird analog zu Phase 1 der TLS Tunnel aufgebaut. Als Entscheidungskriterium für den Server dient zum Beispiel die seit der letzten erfolgreichen Authentifizierung des Clients vergangene Zeit. Im Falle einer Wiederaufnahme der Sitzung wird direkt der Result TLV Austausch eingeleitet (protected termination), also fast die gesamte Phase 2 übersprungen.

Die EAP-FAST Master Session Keys werden wie in 4.9.2 beschrieben abgeleitet. Eine kryptographische Bindung ist überflüssig, da keine inneren EAP Methoden ausgeführt wurden. Es wird dann

S-IMCK$_n$ =SKS

gesetzt und der MSK wie angegeben abgeleitet. Der Vollständigkeit halber ist anzumerken, dass nach Wiederaufnahme einer Sitzung und dem hier geschilderten Ablauf durchaus eine vollständige Phase 2 vom Server eingeleitet werden kann. Ein Wechsel des Netzwerks (bei gleichbleibendem EAP Server) könnte ein Grund hierfür sein.

4.9.5 Undurchsichtigkeit des PAC-Opaque Anteils

Auch nach vollständiger Durchsicht der Spezifikation wird die genaue Absicht, die der PAC Opaque Anteil verfolgt, nicht offensichtlich. Insbesondere genauere Angaben hinsichtlich seines Inhalts sind zu vermissen.

Er kann eine beliebige Länge annehmen und ist serverspezifisch. Das bedeutet, dass außer dem Herausgeber des PAC keine Partei Kenntnis über dessen inhaltliche Zusammensetzung hat. Diese Daten sind für den Client nicht interpretierbar, was die Spezifikation in Abschnitt 12.10.3 relativ deutlich ausdrückt:

> "The PAC-Opaque section contains data that is opaque to the recipient, the peer is not the intended consumer of PAC-Opaque and thus should not attempt to interpret it. A peer that has been issued a PAC-Opaque by a server MUST store that data, and present it back to the server as is, in the PAC-Opaque clientHello extension field. [...] Each client must not parse any PAC-Opaque data given to it."

Im PAC Opaque Anteil ist in geeigneter Weise die Identität des Clients zu verankern. Eine Implementierung wird nämlich dazu angehalten, in EAP-FAST Phase 2 diese mit der Identität zu vergleichen, die der Client in einer inneren EAP Methode bekanntgibt. Da die Präferenz von EAP-FAST eindeutig auf passwortbasierten inneren EAP Methoden liegt, müssten also der Benutzername und das Passwort in geeigneter Weise in den PAC Opaque einfließen. Ein Hinweis auf Seite 9 der Spezifikation bestätigt dies:

> "The PAC-Opaque can only be interpreted by the AS to recover the required information for the server to validate the peer's identity and authentication. For example, the PAC-Opaque may include the PAC-Key and the PACÃs peer identity."

Für interessierte Hersteller, die dieses Verfahren zu implementieren gedenken, dürfte die Undurchsichtigkeit in der Zusammensetzung des PAC Opaque eine Schwierigkeit bedeuten. Da dieser in EAP-FAST Phase 1 unverschlüsselt zum Server geschickt wird, ist er somit für einen Angreifer problemlos abzufangen. Somit erscheint das Konzept noch undurchsichtiger. Entscheidend ist, dass durch dessen Inbesitznahme kein Rückschluss auf den symmetrischen Schlüssel, den PAC Key, möglich sein darf.

Zusammenfassend erscheint das grundsätzliche, aus der ersten Draft Version ersichtliche Konzept, das EAP-FAST verfolgt, zukunftsorientiert und zukunftsweisend zu sein. Dabei verfolgt das Design insbesondere eine Anwendung im Wireless Umfeld. Die als Vorbereitung stattfindende Provisioning Phase dient dazu, ein schwaches Credential, nämlich die Benutzername / Passwort Kombination, in ein starkes Credential, den PAC, zu überführen. Da diese Phase selten ausgeführt wird, wird sie durch asymmetrische kryptographische Verfahren bestimmt. Im Zuge der Authentifizierung wird das zwischen Client und Server geteilte Geheimnis, das Protected Access Credential, herangezogen, welches die Verwendung schneller symmetrischer Kryptographie ermöglicht.

4.10 EAP-SIM

Weltweit sind Telekommunikationsfirmen, insbesondere Betreiber von Handy-Netzen, inbegriffen, die rapide an Bedeutung gewinnende WLAN Technologie in ihr bestehendes Angebot zu integrieren. Ihr entscheidender Vorteil ist die bereits ausgebaute Infrastruktur in Form eines weltumspannenden Netzes und der Besitz von ausgeklügelten AAA-Abrechnungssystemen, den sie zu nutzen suchen.

EAP-SIM ist eine EAP Methode, die auf symmetrischer Kryptographie basiert und die Infrastruktur der GSM Authentifizierung wiederverwendet. Das Verfahren wurde von Cisco und Nokia entwickelt und ist seit seiner ersten Veröffentlichung im Februar 2001 mittlerweile mit Draft Version 13 in einem sehr ausgereiften Stadium. EAP-SIM ist unter der Bezeichnung "Nokia IP smart card authentication" bei der IANA gelistet und hat die Typnummer 18 zugewiesen bekommen.

Zunächst wird eine kompakte Darstellung des Authentifizierungsvorgangs im GSM Netz geliefert und aufgezeigt, welche Rolle das Subscriber Identity Module (SIM) dabei spielt. Kurz eingegangen wird dabei auf die beteiligten Parteien. Erst damit sind Grundlagen soweit gelegt, um auf die Übertragung der GSM Authentifizierung in WLAN einzugehen und den prinzipiellen EAP Nachrichtenaustausch sowie die Generierung des geforderten Schlüsselmaterials vorzustellen.

4.10.1 Authentifizierung in GSM

Jedes mobile GSM Endgerät ist mit einer SIM Karte ausgestattet. Auf einer SIM Karte (Abbildung 4.23) ist letzlich ein Computer im Miniaturformat untergebracht, der Speicher für Daten und Anwendungen, einen Prozessor und eine I/O-Schnittstelle anbietet, um mit dem Gerät bzw. dem Benutzer kommunizieren zu können. Die Größenordnung des Speichers ist 16-64 Kilobyte.

Die SIM Karte enthält

- MSISDN: Mobile Subscriber Integrated Service Digital Network. Öffentliche Rufnummer.
- IMSI: International Mobile Subscriber Identity. Eine weltweit eindeutige Kennung eines Teilnehmers. Sie kann bis zu 15 Ziffern lang sein und ist im Home Location Register (HLR) im Tupel (IMSI, MSISDN) hinterlegt.
- TMSI: Temporary Mobile Subscriber Identity. Temporäre Teilnehmerkennung.

sowie diverse sicherheitsrelevante Angaben

- K_i: Individual Subscriber Authentication Key, ein 128bit langer symmetrischer, geheimer Schlüssel. Er kann nicht direkt von der SIM ausgelesen werden und wird zudem niemals direkt über die Luftschnittstelle übertragen. Das Tupel (IMSI, K_i) steht auch im Home Location Register des Authentication Centers (AuC).
- PIN: Geheimzahl, für die Authentifizierung des Benutzers am mobilen Endgerät.
- Algorithmus A3: Einweg-Funktion, für die Authentifizierung des Teilnehmers nötig.
- Algorithmus A8: Schlüsselstrom-Generator, um den Sitzungsschlüssel K_C zu generieren.

Da die Algorithmen A3 und A8 dieselben Eingabeparameter haben, werden sie oft in einem Algorithmus zusammengefasst. Dessen 96bit Ausgabewert ist dann entsprechend in

32bit SRES und 64bit K_C aufzuteilen. Der Vollständigkeit halber soll der "dritte" Algorithmus genannt werden, der A5 (A5/0, A5/1, A5/2). Er ist ein symmetrischer Blockchiffre, dessen Spezifikation nicht öffentlich ist, sondern nur dessen Schnittstelle. Aufgrund seiner Rechenintensivität ist er direkt in der Hardware des mobilen Endgeräts untergebracht.

Abbildung 4.23: Subscriber Identity Module (SIM)

Der Authentifizierungsvorgang im GSM Netz besteht in einer Überprüfung der SIM des Teilnehmers auf Gültigkeit. Der Ablauf lässt sich wie folgt umreißen: Wünscht ein Teilnehmer Dienste des GSM Netzes in Anspruch zu nehmen (z.B. durch ein Telefonat, Senden einer SMS oder multimedialer Daten), kontaktiert sein mobiles Gerät die Basisstation und übermittelt dieser die von der SIM ausgelesene IMSI oder TMSI. Die Basisstation wiederum nimmt Verbindung mit dem AuC auf und erhält ein GSM-Tripel (RAND,SRES',K_C), das sich aus einem Challenge, der zugehörigen Challenge Response sowie dem Sitzungsschlüssel zusammensetzt.

Der 128bit Challenge RAND wird zum Teilnehmer geschickt, welcher die 32bit lange Challenge Response SRES=A3(RAND,K_i) berechnet und zur Basisstation zurücksendet. Die Basisstation, selbst übrigens zu keinem Zeitpunkt im Besitz geheimer Schlüssel K_i, vergleicht daraufhin SRES' mit SRES auf Übereinstimmung. Ist dies der Fall, ist der Teilnehmer authentifiziert, aber noch nicht zur Kommunikation authorisiert. Hierzu wird der temporäre 64bit Sitzungsschlüssel K_C=A8(RAND,K_i) benötigt, mit dem fortan durch Anwendung des A5-Algorithmus Ver- und Entschlüsselung erfolgt. Diese Berechnung muss nur vom mobilen Gerät des Teilnehmers erfolgen, da die Basisstation K_C bereits durch das GSM-Tripel erhalten hat.

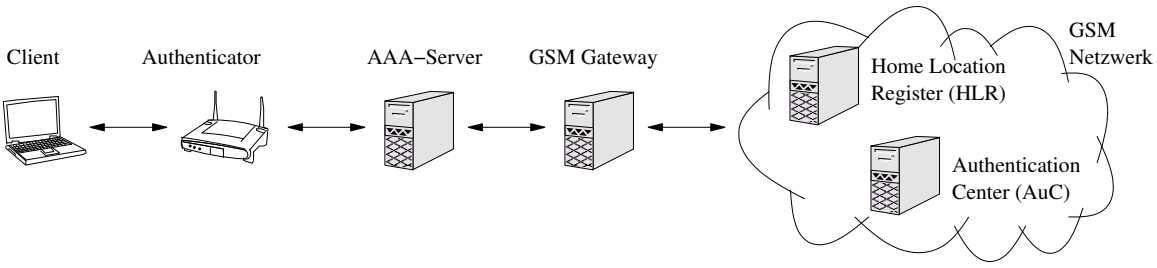


Abbildung 4.24: WLAN und GSM Netzwerk

Der Zusammenhang GSM zu WLAN besteht nun darin, dass auf Grundlage der SIM Karte (bzw. der auf ihr gespeicherten Credentials) die Authentifizierung des Clients erfolgen soll. Unter Client ist ein PDA, ein Notebook oder ein weiteres mobiles Endgerät zu verstehen. Statt eines Zertifikats oder einer Benutzername/Passwort Kombination kommt es also bei EAP-SIM auf Besitz einer (gültigen) SIM Karte an. Wichtig ist also die Feststellung, dass EAP-SIM lediglich der Authentifizierung dient. Der Zugang zum Netz erfolgt über einen Access Point bzw. im Falle eines Wireless Internet Service Providers (WISP) über einen öffentlichen Hotspot.
Wireless LAN und GSM Netz (Abb. 4.24) sind durch einen Gateway verbunden, dessen

technische Einzelheiten hier nicht interessieren, nur die Tatsache, dass der AAA-Server eine sichere Verbindung zum GSM Backbone-Netzwerk, insbesondere dem Home Location Register (HLR) hat.

4.10.2 Protokollbeschreibung

Die GSM-SIM Authentifizierung wurde entsprechend erweitert, um den Anforderungen an eine EAP Methode gerecht zu werden. So ist beispielsweise gegenseitige Authentifizierung und Generierung des geforderten Schlüsselmaterials hinzugekommen. Um die Schlüsselstärke zu erhöhen und um von GSM bekannte Schwächen zu beseitigen, werden bei EAP-SIM mehrere GSM Tripel herangezogen. Die aus den RAND Challenges abgeleiteten Sitzungsschlüssel K_C werden in einem anschließenden Schritt geeignet kombiniert. Die Vorschläge zur Anonymität werden durch Einführung von Pseudonymen aus GSM übernommen. Zur Sicherstellung der Integrität und Authentizität ist ein Message Authentication Code (MAC) eingeführt worden. Ausführlich geht die EAP-SIM Beschreibung weiterhin auf Re-Authentication ein, das stark optimiert ist und äußerst schnelle Wiederaufnahme der Sitzung bei Roaming ermöglicht.

In Abbildung 4.25 wird der Nachrichtenverlauf eine erfolgreichen Authentifizierung dargestellt.

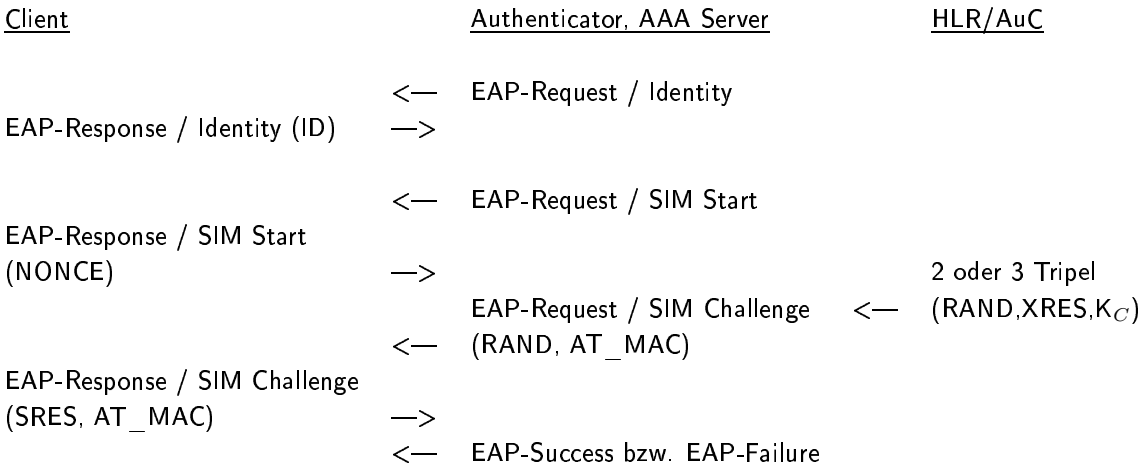


Abbildung 4.25: EAP-SIM Authentifizierung

Der Client kann zwischen zwei Arten von Benutzernamen auswählen, um seine Identität gegenüber dem Netzwerk preiszugeben. Die eine Möglichkeit besteht durch Übermitteln des permanenten Benutzernamens (bei 1123456789098765@myoperator.com ist dies 1123456789098765), die andere durch Schicken eines Pseudonyms. Im Falle von 3s7ah6n9q@myoperator.com ist das Pseudonym gerade 3s7ah6n9q. Das Pseudonym ist äquivalent zur TMSI, aber dennoch ein davon unterschiedlicher Wert, da die TMSI einem Einsatz in der Telefonie vorbehalten ist. In der SIM Start Antwort sendet der Client einen zuvor generierten Zufallswert NONCE mit, der für die Authentizität und den Integritätsschutz der folgenden Nachrichten in die MAC-Funktion eingeht. Die Absicht des MAC ist zum einen, Wiedereinspielungen zu entdecken, zum anderen die Authentizität des Absenders sicherzustellen. Geht ein Vergleich zwischen empfangenen und selbst berechneten MAC negativ aus, wird die Authentifizierung abgebrochen.

Nach Kontaktierung des AuC erhält der Authentication Server von diesem mehrere

GSM Tripel. Die Spezifikation von EAP-SIM bestimmt deren Anzahl auf zwei oder drei, wobei der GSM Standard bis zu fünf Tripel vorsieht. Fortan findet ein Challenge Response Austausch statt. Der vom Authentication Server aus den vorliegenden GSM Tripeln erzeugte Challenge RAND wird zum Client geschickt, der daraus die Antwort SRES=A3(RAND,K_i) berechnet und zurückschickt. Bei Übereinstimmung mit der eigenen ermittelten Challenge Response gilt der Client als erfolgreich authentifiziert und wird mit einer EAP-Success Nachricht über den Ausgang informiert.

Die Sicherheit von EAP-SIM basiert auf der Kenntnis der GSM Tripel (vgl. [Sc03]). Das Netzwerk beweist dem Client seine Echtheit, indem es einen korrekten AT_MAC berechnet, der in der EAP-Request / SIM Challenge Nachricht mitgeführt wird. In den MAC fließen die RAND und K_C Werte aus den GSM Tripeln mit ein. Der Client hat auf die Bildung der Tripel keinen Einfluss und kann zudem ihre Herkunft nicht überprüfen. Kommt ein Angreifer also in den Besitz von zwei (bzw. drei) GSM Tripeln, dann kann er sich gegenüber dem Client als "echtes" Netzwerk ausgeben. Die Spezifikation von EAP-SIM gibt als Quelle gültiger GSM Tripel unter anderem den physikalischen Zugang zur SIM Karte an. Weiterhin kann der Client nicht zwischen bereits benutzten und frischen RAND Challenges unterscheiden, so dass einmal erlangte GSM Tripel wiederverwendet werden können.

4.10.3 Schlüsselableitung

Der Master Key (MK) wird aus den $n \in \{2, 3\}$ Sitzungsschlüsseln K_C, dem NONCE und weiteren Werten wie folgt abgeleitet

$$\text{MK} = \text{SHA-1 (Identity } | \; K_{C1} | \ldots | K_{Cn} | \text{ NONCE } | \text{ Version List } | \text{ Selected Version)}$$

Zur Generierung stärkeren Schlüsselmaterials werden also mehrere, jeweils höchstens 64bit lange GSM Sitzungsschlüssel herangezogen. Es ist erkennbar, dass die Sicherheit von EAP-SIM auf den Sitzungsschlüsseln K_C beruht, welche deswegen unbedingt geheimzuhalten und nicht persistent zu hinterlegen sind. Die Identität ist die aus der IMSI oder dem Pseudonym extrahierte Identitätszeichenkette. Gefolgt von den Schlüsseln K_C, die in der gleichen Reihenfolge wie die RAND Challenges erscheinen. Die Integration des Zufallswerts soll Wiedereinspielungen unterbinden und für die Aktualität des Master Keys garantieren. Die letzten beiden Parameter sind während der Authentifizierung verhandelt worden und stehen für die verwendete EAP-SIM Version sowie eine 2 Byte lange Version List.
Aus diesem Master Key werden durch Anwendung einer PRF (auf NIST FIPS 186-2 aufbauend) nun einerseits kurzlebige Schlüssel für den Schutz der EAP-SIM Nachrichten, andererseits der Master Session Key (MSK) sowie der EMSK abgeleitet, beide in der Länge von 64 Byte. Somit ist die Forderung nach dem zu exportierenden Schlüsselmaterial erfüllt.

Die für einen Zugriff des mobilen Endgeräts auf die SIM Karte nötige Verbindung kann auf mehrere Arten geschehen. Ist die Authentifizierung mit der SIM Karte aus dem Mobiltelefon durchzuführen, so lassen sich beide Geräte durch Bluetooth oder durch ein GPRS Modem zusammenführen. Auf einem der letzten Treffen der 3GPP (3G Partnership Project) Security WG wurde allerdings ein erfolgreicher Man-in-the-middle Angriff für die Bluetooth-Variante vorgeführt ([Ga04]). Alternativ kann eine zweite SIM Karte für den WLAN-Zugang bzw. die nötige Authentifizierung zur Verfügung gestellt werden. Üblicherweise wird diese über einen USB SIM Reader oder einen GPRS PCMCIA Adapter dem mobilen Endgerät direkt zugänglich gemacht wird.

4.11 EAP-AKA

Die Abkürzung "AKA" steht für "Authentication and Key Agreement" Protokoll und ist das UMTS (Universal Mobile Telecommunication System) Äquivalent zum GSM Authentifizierungsprotokoll. Während bei GSM die SIM Karte als Credential dient, ist es bei UMTS die USIM Karte. Diese bietet in der Grundfunktionalität wie die GSM SIM Karte, nämlich Identifikation und Authentifikation des Teilnehmers und Speicherung von System- und Nutzerdaten. Erweitert wurden die Interna, d.h. die Dateistruktur und Zugriffsrechte, sowie die Eigenschaft, multiapplikationsfähig zu sein. Die Entwicklung von EAP-AKA wird von Ericsson und Nokia vorangetrieben und ist ebenfalls wie EAP-SIM schon sehr weit fortgeschritten. Im Mai 2001 zum ersten Mal veröffentlicht ist die aktuelle Spezifikation vom Oktober 2003 in der 11. Überarbeitung angelangt. EAP-AKA macht von der UMTS Authentifizierungs-Infrastruktur Gebrauch.

Über UMTS und dessen Zukunft, insbesondere in Deutschland, kann und soll in diesem Rahmen nicht spekuliert werden. Gleichwohl ist aber eine technische Eigenschaft von UMTS hervorzuheben. Das Ziel von UMTS ist, das für Sprachübermittlung geschaffene und für komplexe Datenübertragungen zu langsame GSM abzulösen, da UMTS einen Datendurchsatz bis zu 2MBit/s ermöglicht (Tabelle 4.3). Dabei ist der Datendurchsatz von der Bewegungsgeschwindigkeit des Clients abhängig, so dass im ungünstigsten Fall nur etwas mehr als die doppelte ISDN-Geschwindigkeit zur Verfügung steht. Das reservierte Frequenzband von 1990 MHz bis 2200 MHz liegt direkt unterhalb von WLAN 802.11b/g und weist somit ähnliche physikalische Eigenschaften auf (Reichweite, Datendurchsatzrate usw.), wobei aber andere Modulationsverfahren zum Tragen kommen.

	Unkomprimiert
Bitrate	9.6 kByte/s

	bis 10 km/h	bis 120 km/h	bis 500 km/h
Bitrate	bis 2 Mbit/s	384 KBit/s	144 KBit/s

Tabelle 4.3: GSM vs. UMTS Übertragungsgeschwindigkeiten

Es laufen große Anstrengungen und Überlegungen bei Mobilgeräteherstellern und Netzbetreibern, wie Internet, WLAN und Telefonie in Einklang zu bringen sind. Auch über eine Konvergenz der Netze wird heftig diskutiert. All das sollte zu einem späteren Zeitpunkt, wenn die Technik hinreichend entwickelt und der Markt hinreichend Bereitschaft zeigt, genauer betrachtet werden.

4.12 EAP-SPEKE

SPEKE steht abkürzend für "Simple Password Exponential Key Exchange" und ist ein Vertreter der Starken-Passwort Methoden. Basierend auf dem Zero Knowledge Passwortproblem, dass also zwei Seiten sich gegenseitig den Besitz des Passworts beweisen, ohne es direkt auszutauschen, ermöglicht EAP-SPEKE die Verwendung von einfach zu erinnernden und durchaus auch kurzen, also benutzerfreundlichen Passwörtern. Dabei wird durch kryptographische Raffinesse die übliche Anfälligkeit passwortbasierter Authentifizierungsverfahren für Wörterbuchangriffe umgangen. Starke-Passwort Methoden zeichnen sich aus durch

- stabile Sicherheit auch bei sehr kurzen Passwörtern
- kein Bedarf an weiteren Schlüsseln, insbesondere Zertifikaten

- gegenseitige Authentifizierung
- Sicherheit gegen Lauschen und offline stattfindende Brute-Force Angriffe
- Generieren eines Sitzungsschlüssels.

EAP-SPEKE verwendet nur im anfänglichen Diffie-Hellman Austausch schwergewichtige Berechnungen. Danach kann symmetrische Kryptographie erfolgen und somit ist EAP-SPEKE sehr gut im Wireless Umfeld einsetzbar, was die Firma Interlink Networks erkannt und das Verfahren im Februar 2003 exklusiv durch Erwerb einer Lizenz von der Entwicklerfirma Phoenix Technologies gesichert hat. Solche sogenannten "IPR-encumbered methods"[8] erfreuen sich weit weniger Beliebheit als frei verfügbare Verfahren. Dennoch soll kurz das Prinzip deutlich gemacht werden.

Nach Erhalten der Identität generiert der Server eine große Zufallszahl b und besorgt sich, zum Beispiel aus einer Datenbank, das Passwort p des Clients, das in eine primitiven Wurzel modulo m überführt wird. Der Modulus sowie $B = p^{2b} \mod m$ werden dem Client zugestellt. Dieser berechnet $A = p^{2a} \mod m$, den Master Session Key (MSK) $K = B^a \mod m$ sowie Proof_{AK}=Hash("A"|A|K) und schickt Proof_{AK} zum Server, der daraus den Master Session Key $K = A^b \mod m$ ableitet. Mit Wissen von K kann er nun die Richtigkeit von Proof_{AK} prüfen, nämlich durch eigene Berechnung von Hash("A"|A|K) und Vergleich mit Proof_{AK}.
Um dem Client zu zeigen, dessen Passwort p zu kennen, berechnet der Server Proof_{BK}=Hash("B"|B|K) und schickt diesen Beweis an den den Client. Da dem Client alle Hashparameter bekannt sind, ist der empfangene Proof_{BK} mit dem eigenen berechneten Wert zu vergleichen. Den Ausgang des Vergleichs teilt er dann dem Server mit.

Beide Seiten haben also den Master Session Key K unabhängig voneinander berechnet. Allein mit Kenntnis von A oder B kann auf das Passwort p kein Rückschluss gezogen werden. Der gegenseitige Wissensbeweis wird durch folgende Gleichung offensichtlich:

$$B^a \mod m = p^{2b \cdot a} \mod m = A^b \mod m = p^{2a \cdot b} \mod m$$

C—>S: ID
S—>C: $(B = p^{2b} \mod m, m)$
C—>S: (A, Proof_{AK})
S—>C: Proof_{BK}
C—>S: Success / keine Antwort
S—>C: Success / Failure

Abbildung 4.26: Schematischer Nachrichtenaustausch bei EAP-SPEKE

Diese Starke-Passwort Methoden gehen auf Steve Bellovin und Michael Merritt (beide von Bell Labs) zurück, die die Familie der EKE (Encrypted Key Exchange) ins Leben gerufen haben. Die Methode SPEKE stammt von David Jablon.
Für weitere Informationen zu diesem Thema ist am besten von der SPEKE Webseite ([Ph03]) oder von David Jablons Homepage `http://world.std.com/~dpj/` aus zu navigieren. Die Arbeitsgruppe P1363.2 der IEEE, "Password-Based Public-Key Cryptography" beschäftigt sich im großen Rahmen mit dieser Thematik. Ihre Webseite `http://grouper.ieee.org/groups/1363/passwdPK/` stellt ebenfalls eine Ausgangsbasis dar.

[8] IPR = Intellectual Property Right (Schutz des geistigen Eigentums); encumbered = vorbelastet

Kapitel 5

Zusammenfassung

In dieser Arbeit ist die mit Einführung des WLAN Standards IEEE 802.11i zur Verfügung stehende verbesserte Authentifizierung eingehend beschrieben worden. Nach einer intensiven Auseinandersetzung mit den beteiligten Protokollen 802.1x, EAP und RADIUS sind mehrere aktuelle EAP Methoden herausgegriffen und deren grundlegenden Eigenschaften wie charakteristischen Besonderheiten herausgearbeitet worden. Bei der Beschäftigung mit dieser Thematik wird deutlich, es mit einem äußerst komplexen Gebiet zu tun zu haben, in dem viele Techniken, Protokolle und Spezifikationen ineinandergreifen bzw. aufeinander aufbauen.
Zudem musste das Zusammenspiel der unterschiedlichen Akteure durchschaut werden. Die IEEE und IETF legen Grundlagen und schaffen ein Rahmenwerk - ihr Aufgabenbereich endet mit der Spezifikation von Schnittstellen. Dass dies bereits eine schwierige, äußerst zeitintensive und an Schaffenskraft und Kreativität hohe Anforderung stellende Aufgabe ist, wird spätestens durch einen Blick in die EAP WG Mailing List offensichtlich. Die Entwicklung von konkreten EAP Authentifizierungsmethoden liegt hingegen in den Händen anderer: Es sind fast ausschließlich Firmen, vor allem Branchenriesen, die jeweils mehrere Mitarbeiter mit dieser Aufgabe betreut haben. Die hierein erfolgte Investition soll sich letzlich auch in deren Produkten, z.B. Software oder Geräten, wiederfinden, und somit ist die Motivation in die Entwicklung neuer Verfahren insbesondere durch das Versprechen wirtschaftlicher Vorteile begründet.

	EAP-MD5	LEAP	EAP-SPEKE	EAP-TLS
Erstes Erscheinen	1998	2000	Feb. 2002	Okt. 1999
Entwicklung durch	RFC2284	Cisco	Phoenix Tech.	Microsoft
Authentifizierung des Servers	-	Passwort	Passwort	Zertifikat
Authentifizierung des Clients	Passwort	Passwort	Passwort	Zertifikat
WLAN geeignet?	nein	nein	ja	ja
Einrichtungsaufwand	gering	gering	gering	sehr hoch
Sicherheit des Credentials	schwach	schwach	N/A	stark
Spezifikation	öffentlich	proprietär	N/A	öffentlich
Entwicklungsstadium	abgeschlossen	N/A	abgeschlossen	abgeschlossen

Tabelle 5.1: EAP-Methoden Zusammenfassung (Teil 1)

	PEAP(v2)	EAP-TTLS	EAP-FAST	EAP-SIM
Erstes Erscheinen	Aug. 2001	Aug. 2001	Feb. 2004	Feb. 2001
Entwicklung durch	Cisco, MS, RSA	Funk, Certicom	Cisco	Cisco, Nokia
Authentifizierung des Servers	Zertifikat	Zertifikat	Zertifikat	GSM-Tripel
Authentifizierung des Clients	EAP (beliebig)	EAP (beliebig)	EAP (bel.), Legacy	GSM-SIM
WLAN geeignet?	ja	ja	ja	N/A
Einrichtungsaufwand	mittel	mittel	gering	gering
Sicherheit des Credentials	stark	stark	stark	mittel
Spezifikation	öffentlich	öffentlich	öffentlich	öffentlich
Entwicklungsstadium	Draft	Draft	Draft	Draft

Tabelle 5.2: EAP-Methoden Zusammenfassung (Teil 2)

Zum Abschluss dieser Arbeit soll, soweit es möglich ist, ein Resümee gezogen und der Versuch unternommen werden, die sich abzeichnenden Tendenzen und Richtungen der Entwicklung zu skizzieren. Die in dieser Arbeit vorgestellten EAP Methoden sind in den Tabellen 5.1 und 5.2 einander gegenübergestellt. Die Abkürzung N/A wird verwendet, wenn eine Antwort nicht bekannt ist.

Als sich abzeichnender Trend ist die Verfolgung der Ziele

- geringer Einrichtungsaufwand
- Einfachheit für den Benutzer
- hohe Flexibilität
- gute Skalierung

mit dem Bieten größtmöglicher Sicherheit anzusehen.

Der erforderliche Aufbau einer Public Key Infrastruktur, wie sie EAP-TLS erfordert, ist sehr komplex und für die meisten Anwendungsszenarien zu aufwändig. Mit der Erstellung von Zertifikaten, einer durchaus realisierbaren Unternehmung, ist einer PKI noch nicht hinreichend Sorge getragen. Die Bemühungen müssen sich weiterhin richten auf Verteilung, Erneuerung und Zurückziehen von Zertifikaten. Entweder sind diese auf dem Gerät des Benutzers oder auf einem portablen Speichermedium (z.B. USB Stick) zu hinterlegen. Ein Verlust des Geräts bzw. des Speichermediums hat in der Regel unmittelbar den Zugang Unberechtigter zur Folge.

Zukunftsweisender sind demzufolge EAP Methoden, die nicht auf client- und serverseitig verteilte Zertifikate bauen. Durch hinreichend gute Erfahrungen und kryptographische Untersuchung unterstützt erscheinen die Verfahren günstiger, die einen TLS Tunnel als sicheren Kommunikationskanal aufbauen und durch diesen flexibel weitere Authentifizierungsprotokolle erfolgen lassen. Die Flexibilität hinsichtlich der Auswahl innerer Methoden ist ebenfalls nicht unbedeutend. Zu der Kategorie "Getunnelte EAP Methoden" gehören PEAP, EAP-TTLS und EAP-FAST.

Starke Anstrengungen werden zur Zeit bei den Preshared Keys Methoden unternommen (vgl. [18]), mit dem Ziel, das überholte, aber leider standardisierte EAP-MD5 abzulösen. Zu dieser Kategorie von Methoden gehören EAP-SPEKE, EAP-PSK, SRP-SHA1 (Secure Remote Password), EAP-Archie und EAP-SKE (Shared Key Exchange). Shared Key Verfahren bringen den großen Vorteil, für die während der Authentifizierung erforderliche

Verschlüsselung symmetrische Kryptographie zu verwenden. Insbesondere im Hinblick auf den Einsatz im Wireless Bereich sind diese Verfahren interessant, da sie den rechenschwachen Geräten entgegenkommen. Es ist allgemein der Trend zu beobachten, dass der Wireless Anwendungsbereich mit in die Draft Updates eingeflossen ist - man ist sich also durchaus der Zielgruppe, allem voran IEEE 802.11i WLAN, bewusst. Das Design wird hierdurch maßgeblich mitbestimmt.

Die Phase der Authentifizierung in IEEE 802.11i nimmt eine Schlüsselstellung ein. Ihr Ausgang bedingt die Zugangs- und Zugriffskontrolle zum Netzwerk und zum Nutzen der angebotenen Ressourcen, so dass die Untersuchung der verfügbaren Verfahren, insbesondere ihrer Protokolle und kryptographischen Sicherungsmechanismen ein hoher Stellenwert zukommen muss. Ein Ende der Serie von Innovationen und technischen Überraschungen ist in diesem Bereich mit Sicherheit auszuschließen.

Literaturverzeichnis

[802.11 ML] IEEE 802.11 Working Group, Mailing List.
http://grouper.ieee.org/groups/802/11/private/e-mail-archive-normal/

[Ab02] Aboba, Bernard.
IEEE 802.1X Pre-Authentication. Juni 2002.
http://www.drizzle.com/~aboba/IEEE/11-02-TBDr0-I-Pre-Authentication.doc

[Ar01] Arbaugh, William. Mishra, Arunesh.
An Initial Analysis of the 802.1x Standard. University of Maryland. Februar 2002.
http://www.cs.umd.edu/~waa/1x.pdf

[Ag02] Agere Systems.
Orinoco Wireless LAN Security. *Kein Erscheinungsdatum.*
Response to "An Initial Security Analysis of the IEEE 802.1X Standard".
http://domino.mms.de/tech.nsf/0/f894f0b9d7338398c1256914004319c0/$FILE/Wireless_LAN%20_Security_Response.pdf

[Ar03] Arbaugh, William. Petroni, Nick.
The Dangers of Mitigating Security Design Flaws: A Wireless Case Study.
2003, IEEE Security & Privacy.

[ArWi] Arbaugh, William.
Wireless Research. 802.11 Security Vulnerabilities. A History.
http://www.cs.umd.edu/~waa/wireless.html

[As01] Asokan, N. Niemi, Valtteri. Nyberg, Kaisa.
Man-in-the-Middle in Tunnelled Authentication Protocols.
Extended Abstract. Nokia Research Center. April 2003.
http://www.saunalahti.fi/~asokan/research/tunnel_extab.pdf

[BW-ML] Bawug Wireless Mailing List.
http://lists.bawug.org/pipermail/wireless/

[Bo01] Borisov, Nikita. Goldberg, Ian. Wagner, David.
Intercepting Mobile Communications: The Insecurity of 802.11.
University of California, Berkeley. 2001.
http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf

[Br03] Brewin, Bob.
Cisco develops WLAN security protocol to defeat password attacks.
The fix will be available by the end of March. Computerworld, 12.02.04.
http://www.computerworld.com/securitytopics/security/story/0,10801,90163,00.html

[Ci01a] Roshan, Pejman.
An In-Depth Look at 802.11 Wireless LAN Security and the Cisco Wireless Security Suite.
Cisco Internal Use Only. Cisco Systems, 2001.
http://www.comstor.com/wireless/pdfs/WNBU_Security_White_Paper.pdf

[Ci02a] Cisco Systems.
A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite. Whitepaper, 2002.
http://www.cisco.com/warp/public/cc/pd/witc/ao1200ap/prodlit/wswpf_wp.pdf

[Ci02b] Cisco Systems.
Cisco Aironet Response to University of Maryland's Paper, "An Initial Security Analysis of the IEEE 802.1x Standard". August 2002.
http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/1680_pp.htm

[Ci03a] Cisco Systems.
SAFE: Wireless LAN Security in Depth - version 2; Whitepaper. Oktober 2003.
http://www.cisco.com/warp/public/cc/so/cuso/epso/sqfr/safwl_wp.htm

[Ci03b] Cisco Systems.
Security notice for Dictionary Attacks on Cisco LEAP. Document ID: 44281. 03.08.2003.
http://www.cisco.com/warp/public/707/cisco-sn-20030802-leap.shtml

[Ci03c] Cisco Systems.
Cisco Response to Dictionary Attacks on Cisco LEAP. Product Bulletin No. 2331. November 2003.
http://www.cisco.com/warp/public/cc/pd/witc/ao350ap/prodlit/2331_pp.htm

[Ci03d] Cisco Systems.
Cisco - LEAP and RADIUS. 2003.
http://www.cisco.com/warp/public/102/wlan/LEAPRADIUS.pdf

[Ci03e] Ahlawat, Sharad.
Re: Weakness in LEAP Challenge/Response.
Beitrag von 07.10.2003.
http://cert.uni-stuttgart.de/archive/bugtraq/2003/10/msg00108.html

[Ci04a] Cisco Systems.
Cisco Aironet 1200 Series. Frequently Asked Questions, EAP-FAST. März 2004.
http://www.cisco.com/warp/public/cc/pd/witc/ao1200ap/prodlit/eapfs_qa.pdf

[Co04] Courtois, Nicolas.
Is AES a Secure Cipher?
http://www.cryptosystem.net/aes/
http://www.minrank.org/~courtois/myresearch.html

[Cw03] Cam-Winget, Nancy et al.
Security Flaws in 802.11 Data Link Protocols.
Communications of the ACM. Mai 2003.
http://www.cs.berkeley.edu/~daw/papers/wireless-cacm.pdf

[Di04] Ding, Ping. Holliday, Jo Anne et al.
Improving the Security of Wireless LANs By Managing 802.1x Disassociation.
Santa Clara University, California. 2004.
http://www.cse.scu.edu/~jholliday/CCNC04.pdf

[Dict] Wörterbücher (mehrsprachig). ftp://coast.cs.purdue.edu/pub/dict/

[EAP-ML] IETF EAP Working Group Mailing List.
http://mail.frascone.com/pipermail/public/eap/

[EAP11] EAP Working Group. Meeting November 2002.
http://www.ietf.org/proceedings/02nov/136.htm

[Ea03] Eaton, Dennis.
Diving into the 802.11i Spec: A Tutorial
http://www.commsdesign.com/design_library/cd/hn/OEG20021126S0003

[Eis01] Eisinger, Jochen.
Exploiting known security holes in Microsoft's PPTP Authentication Extensions (MS-CHAPv2).
Universität Freiburg, 2001.
http://mopo.informatik.uni-freiburg.de/pptp_mschapv2/

[Er04] Eronen, Pasi. Arkko, Jari.
Authentication components: Engineering experiences and guidelines. Februar 2004.
http://www.arkko.com/publications/tunnelling.pdf

[Evol] evol@ruiner.halo.nu.
Cisco LEAP Insecurities + POC; LEAP Exploit Code. Oktober 2003.
http://cert.uni-stuttgart.de/archive/bugtraq/2003/10/msg00057.html

[Fl01] Fluhrer, Scott; Mantin, Itsik; Shamir, Adi.
Weaknesses in the Key Scheduling Algorithm of RC4; 2001
http://www.crypto.com/papers/others/rc4_ksaproc.ps

[Fu02] Funk Software.
Comments on "An Initial Security Analysis of the IEEE 802.1X Standard", März 2002.
http://www.funk.com/radius/Solns/umdresp_wp.asp

[Ga04] Gauthier, Eric.
A man-in-the-middle attack using Bluetooth in a WLAN interworking environment.
3GPP TSG SA WG3 Security#32. Februar 2004.
http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_32_Edinburgh/Docs/PDF/S3-040049.pdf sowie S3-040047.pdf

[Ha03] Hagedorn, Axel.
IEEE 802.11i Sicherheit in drahtlosen lokalen Netzen.
Diplomarbeit, TU Darmstadt, 2003
http://www.informatik.tu-darmstadt.de/ftp/pub/TI/reports/AxelHagedorn.diplom.pdf

[Hi01] Hill, Joshua.
An Analysis of the RADIUS Authentication Protocol.
InfoGard Laboratories, 2001.
http://www.untruth.org/~josh/security/radius

[Ho03] Hole, Kjell.
Indoor WLAN Design. Part VIII: RADIUS, TLS and Known WPA Attacks. 2003.
http://www.kjhole.com/Standards/WiFi/WiFi-PDF/WLAN8alt.pdf

[In03] Interlink Networks.
EAP Methods for Wireless Authentication, Whitepaper. April 2003.
http://www.interlinknetworks.com/images/resource/EAP_Methods_for_Wireless.pdf

[JNw03] Interlink Networks.
Selecting an EAP Method: Access Point Considerations. Tech Tutorial.
http://www.interlinknetworks.com/news/newsletters/20030810/newsletter.html

[Jtl03] Intel Blueprint.
Wireless LAN (WLAN) End-to-End Guidelines for Enterprises and Public Hotspot Service Providers. November 2003.
http://www.intel.com/business/bss/infrastructure/wireless/deployment/e2e_wlan.pdf

[Jtl00] Weatherspoon, Sultan.
Network Communications Group, Intel Corporation
Overview of IEEE 802.11b Security. Intel Technology Journal, Q2, 2000.
http://developer.intel.com/technology/itj/q22000/articles/art_5.htm

[Ke01] Kerry, Stuart.
WLAN/ Response of WEP Security. 12.02.01
[EAP-ML] msg00901.html

[Mee03] Burns, Jim.
Selecting an appropriate EAP method for your wireless LAN.
Meetinghouse Data Communications, Feb. 2003, Whitepaper

[Mo02] Moskowitz, Robert.
Time to Smarten Up About Security. Juni 2002.
http://www.networkcomputing.com/1313/1313colmoskowitz.html

[Ni01] National Institute of Standards and Technology (NIST).
AES Algorithm (Rijndael) Information.
http://csrc.nist.gov/CryptoToolkit/aes/rijndael/

[Ni02] National Institute of Standards and Technology (NIST).
ed. Karygiannis, Tom. Owens, Les.
Wireless Network Security, 802.11, Bluetooth and Handheld Devices.
NIST SP 800-48, Nov. 2002.
http://csrc.nist.gov/publications/nistpubs/800-48/NIST_SP_800-48.pdf

[Ph03] Phoenix Technologies.
SPEKE. Simple Password-authenticated Exponential Key Exchange.
http://www.integritysciences.com/speke.html

[Pu02] Pulse Inc. What is 802.11 & 802.11b?
http://www.pulsewan.com/data101/802_11_b_basics.htm

[Ri01] Rijmen, Vincent.
The block cipher Rijndael. Homepage.
http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

[Rv01] Rivest, Ron.
WLAN/ Ron Rivest's comments on RC4/WEP issue.
http://grouper.ieee.org/groups/802/11/private/e-mail-archive-normal/
msg01405.html

[Sc03] 3GPP.
EAP support in smartcards and security requirements in WLAN authentication.
TSG SA WG3 Security S3#28, S3-030198. Mai 2003.
http://www.3gpp.org/ftp/tsg_sa/WG3_Security/2003_meetings/TSGS3_28_
Berlin/Docs/PDF/S3-030198.pdf

[Sch98] Schneier, Bruce. Mudge (mudge@L0pht.com).
Cryptanalysis of Microsoft's Point-to-Point Tunneling Protocol (PPTP). 1998.
http://www.schneier.com/paper-pptp.html

[Sch99] Schneier, Bruce.
Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2). 1999.
http://www.schneier.com/paper-pptpv2.html

[Sp00] Specht, Raimund.
Seminar "Mobile Netzwerke" - "Sicherheit in mobilen Netzen".
Seminarausarbeitung, Universität Ulm. 1999.
http://www.spemaus.de/studium/mn/ausarbeitung.xhtml

[St01] Adam Stubblefield, John Ioannidis, Aviel D. Rubin.
Using the Fluhrer, Mantin, and Shamir Attack to Break WEP, August 2001
http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/
stubbl.pdf

[Thc] THC LEAP Cracker Tool suite.
Tools to break the NTChallengeResponse encryption technique e.g. used by Cisco Wireless LEAP Authentication.
http://www.thc.org/releases.php

[Ve02] Ventura, Hakan.
Diameter - Next generations AAA protocol. Student thesis. 2002.
http://www.ep.liu.se/exjobb/isy/2001/3232/exjobb.pdf

[Wa00] Walker, Jesse.
IEEE P802.11 Wireless LANs Unsafe at any key size; An analysis of the WEP encapsulation. Oktober 2000

[Wa02] Walker, Jesse. WEP, TKIP. AES. (3 Websites)
http://www.intel.com/cd/ids/developer/asmo-na/eng/technologies/security/{17780,17769,17770}.htm

[Wa00] Ferguson, Niels et al.
Improved Cryptanalysis of Rijndael. 2000.
http://www.cs.berkeley.edu/~daw/papers/rijndael-fse00.ps

[Wo03] Wong, Stanley.
The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards.
SANS Institute, Mai 2003.
http://www.sans.org/rr/papers/68/1109.pdf

[Wr03a] Wright, Joshua.
README Datei zu asleep, LEAP Exploit Code. Version 1.5, Januar 2004.

[Wr03b] Wright, Joshua.
Weakness in LEAP Challenge/Response.
Beitrag in Security Bugtraq MailingList vom 06.10.2003.
http://cert.uni-stuttgart.de/archive/bugtraq/2003/10/msg00076.html

[Wr03c] Wright, Joshua.
Weaknesses in LEAP Challenge/Response. Präsentation.
http://home.jwu.edu/jwright/presentations/asleap-defcon.pdf

Internet-Drafts

[1] Jose Puthenkulam, Victor Lortz, Ashwin Palekar, Dan Simon.
The Compound Authentication Binding Problem.
Internet-Draft. draft-puthenkulam-eap-binding-04.txt

[2] B. Aboba, D. Simon, J. Arkko, H. Levkowetz.
EAP Key Management Framework.
Internet-Draft. Oktober 2003. draft-ietf-eap-keying-01.txt

[3] Dorothy Stanley. Jesse Walker. Bernard Aboba.
EAP Method Requirements for Wireless LANs.
Internet-Draft. März 2004. draft-walker-ieee802-req-01.txt

[4] N.Cam-Winget, D. McGrew, J. Salowey, H.Zhou
Cisco EAP Flexible Authentication via Secure Tunneling (EAP-FAST).
Internet-Draft. Februar 2004. draft-cam-winget-eap-fast-00.txt

[5] T. Clancy, N. Petroni, W. Arbaugh.
Technique for Method-Specific Fast EAP Rekeying.
Internet-Draft. Februar 2004. draft-clancy-eap-rekeying-00.txt

[6] P. Eronen, T. Hiller, G. Zorn.
Diameter Extensible Authentication Protocol (EAP) Application.
Internet-Draft. Februar 2004. draft-ietf-aaa-eap-04.txt

[7] A. Palekar, D. Simon, G. Zorn, J. Salowey, H. Zhou, S. Josefsson.
Protected EAP Protocol (PEAP) Version 2.
Internet-Draft. Oktober 2003. draft-josefsson-pppext-eap-tls-eap-07.txt

[8] V. Kamath, A. Palekar, M. Wodrich.
Microsoft's PEAP version 0 (Implementation in Windows XP SP1).
Internet-Draft. Oktober 2002. draft-kamath-pppext-peapv0-00.txt

[9] J. Salowey, P. Eronen.
Guidelines for using the EAP Extended Master Session Key (EMSK) .
Internet-Draft. November 2003. draft-salowey-eap-key-deriv-02.txt

[10] H. Tschofenig, D. Kroeselberg, Y. Ohba
EAP IKEv2 Method (EAP-IKEv2).
Internet-Draft. Februar 2004. draft-tschofenig-eap-ikev2-03.txt

[11] P. Funk, S. Blake-Wilson.
EAP Tunneled TLS Authentication Protocol (EAP-TTLS).
Internet-Draft. Oktober 2003. draft-ietf-pppext-eap-ttls-03.txt

[12] J. Arkko, H. Haverinen.
EAP AKA Authentication.
Internet-Draft. Oktober 2003. draft-arkko-pppext-eap-aka-11.txt

[13] H. Haverinen. J. Salowey.
EAP SIM Authentication.
Internet-Draft. Oktober 2003. draft-haverinen-pppext-eap-sim-12.txt

[14] H. Kim. H. Afifi. M. Hayashi.
EAP Bluetooth Application.
Internet-Draft. Februar 2004. draft-kim-eap-bluetooth-00.txt

[15] D. Jablon.
The SPEKE Password-Based Key Agreement Methods
Internet-Draft. Oktober 2003. draft-jablon-speke-02.txt

[16] P. Gutmann.
Use of Shared Keys in the TLS Protocol.
Internet-Draft. Oktober 2003. draft-ietf-tls-sharedkeys-02.txt

[17] B. Aboba.
EAP IANA Considerations.
Internet-Draft. Oktober 2002. draft-aboba-pppext-eap-iana-02.txt

[18] F. Bersani.
EAP shared key methods: a tentative synthesis of those proposed so far.
Internet-Draft. April 2004. draft-bersani-eap-synthesis-sharedkeymethods-00.txt

Request-For-Comments (RFCs)

[RFC2284-] Extensible Authentication Protocol (EAP). Februar 2004. Proposed Standard.
draft-ietf-eap-rfc2284bis-09.txt

[RFC2716] PPP EAP TLS Authentication Protocol. Oktober 1999. Experimental Standard

[RFC2246] The TLS Protocol Version 1.0. January 1999. Proposed Standard

[RFC3546] Transport Layer Security (TLS) Extensions. Juni 2003. Proposed Standard

[RFC3579] RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. Informational

[RFC2548] Microsoft Vendor-specific RADIUS Attributes March 1999 Informational

[RFC2759] Microsoft PPP CHAP Extensions, Version 2. Januar 2000. Informational

IEEE (Draft) Standards

[802.1x] IEEE Std 802.1X-2001, June 2001.
IEEE Standards for Local and Metropolitan Area Networks.
Port based Network Access Control

[802.1x-9] IEEE P802.1X-REV/D9, January 26, 2004.
DRAFT Standard for Local and Metropolitan Area Networks.
Port-Based Network Access Control (Revision)

[802.11i] IEEE P802.11i/D9.0, March 2004.
Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications,
Amendment 6: Medium Access Control (MAC) Security Enhancements

[802.11-99] ANSI/IEEE Std 802.11, 1999 Edition.
IEEE Standards for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements.
Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications
`http://standards.ieee.org/getieee802/download/802.11-1999.pdf`

Anhang

Anhang A - EAP-TLS Paketdump

In diesem Anhang ist eine vollständige EAP-TLS Authentifizierung zu finden. Die Testumgebung besteht aus einem Notebook, einem Linksys 802.11g Access Point und einem PC. Der gewählte Access Point erfreut sich in fortgeschrittenen Anwenderkreisen großer Beliebtheit, da dessen Firmware relativ unkompliziert per TFTP durch eigene Kompilationen ausgetauscht werden kann. Mittlerweile ist der Funktionsumfang der Konfiguration um ein wesentliches höher als in der Originalfirmware von Linksys und beinhaltet unter anderem Dienste wie SSH, FTP und Funktionen wie Traffic Control.

	Supplicant	**Authenticator**	**Authentication Server**
System	Notebook	Access Point	RADIUS Server
OS	Fedora Linux	Linux	Fedora Linux
Hardware	Netgear WG511T	Linksys WRT54G	10/100MBit Ethernet NIC
Software	Xsupplicant	Sveasoft 2.00.8.6sv	Freeradius 0.9.3
Netzwerkgerät	ath0	-	eth1
MAC Adresse	00:09:5B:93:01:08	00:06:25:D7:B4:B3	00:E0:18:89:2D:16
IP-Adresse	192.168.1.2	192.168.1.1	192.168.1.3
ESSID	linksys		
Channel	6		

Abbildung 5.1: EAP-TLS Testumgebung

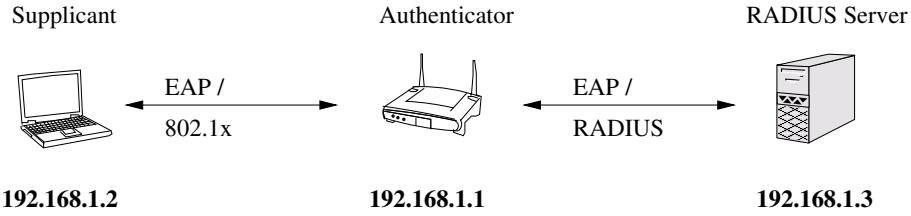
Die EAP-Verhandlung erfolgt auf Clientseite durch XSupplicant, einem Teil des Open1X-Projekts (`http://www.open1x.org/`), auf Serverseite durch Freeradius (`http://www.freeradius.org/`), die beide frei verfügbar sind. Die Konfiguration beispielsweise auf dem Notebook ist denkbar einfach, da sie lediglich in der Anpassung einer Konfigurationsdatei besteht:

`/etc/1x/1x.conf`	Bedeutung
`linksys : id = thomas`	Benutzeridentität
`linksys : cert = <Pfad>/newcert.pem`	X509v3 Clientzertifikat
`linksys : key = <Pfad>/newreq.pem`	Privater Schlüssel
`linksys : root = <Pfad>/cacert.pem`	CA Zertifikat
`linksys : auth = EAP`	EAP Authentifizierung
`linksys : pref = tls`	Methode: EAP-TLS
`linksys : password = kennwort`	Passwort zum privaten Schlüssel
...	...

Für die Erstellung der Zertifikate eignet sich unter Linux die OpenSSL Bibliothek. In diesem Zusammenhang sowie in der Konfiguration von Freeradius sind folgende Tutorials sehr hilfreich und sollten zum Einstieg dienen.

- Ken Roser. Howto: EAP/TLS Setup for FreeRADIUS and Windows XP Supplicant; Version 1.0.4 vom 07.02.03. `http://3w.denobula.com:50000/EAPTLS.html`
- Raymond McKay. FreeRADIUS EAP/TLS - WinXP HOWTO. Version 1.2 10/30/2002. `http://www.impossiblereflex.com/8021x/eap-tls-HOWTO.htm`
- Adam Sulmicki. HOWTO on EAP/TLS authentication between FreeRADIUS and XSupplicant. `http://www.missl.cs.umd.edu/wireless/eaptls/`

In den folgenden beiden Abbildungen ist der Nachrichtenaustausch zwischen Notebook und Access Point sowie zwischen Access Point und RADIUS Server im Überblick dargestellt. Der Paketdump stammt vom Paketsniffer "Ethereal". In Abbildung 5.2 ist die Laufzeit interessant, dass mit konventioneller Hardware eine EAP-TLS Authentifizierung ungefähr eine Sekunde dauert. Der die nächsten Seiten bestimmende Paketdump ist teilweise unter Verweis gekürzt (z.B. Zertifikataustausch) oder auch mit Kommentaren versehen.



1	0.000000	Netgear_93:01:08	LinksysG_d7:b4:b5	EAPOL	Start
2	0.000958	LinksysG_d7:b4:b5	Netgear_93:01:08	EAP	Request, Identity [RFC2284]
3	0.260064	Netgear_93:01:08	LinksysG_d7:b4:b5	EAP	Response, Identity [RFC2284]
4	0.264544	LinksysG_d7:b4:b5	Netgear_93:01:08	EAP	Request, EAP-TLS [RFC2716] [Aboba]
5	0.337028	Netgear_93:01:08	LinksysG_d7:b4:b5	TLS	Client Hello
6	0.347263	LinksysG_d7:b4:b5	Netgear_93:01:08	EAP	Request, EAP-TLS [RFC2716] [Aboba]
7	0.600117	Netgear_93:01:08	LinksysG_d7:b4:b5	EAP	Response, EAP-TLS [RFC2716] [Aboba]
8	0.631545	LinksysG_d7:b4:b5	Netgear_93:01:08	TLS	Server Hello, Certificate, Certificate Request, Server Hello Done
9	0.777759	Netgear_93:01:08	LinksysG_d7:b4:b5	EAP	Response, EAP-TLS [RFC2716] [Aboba]
10	0.786827	LinksysG_d7:b4:b5	Netgear_93:01:08	EAP	Request, EAP-TLS [RFC2716] [Aboba]
11	1.040095	Netgear_93:01:08	LinksysG_d7:b4:b5	TLS	Certificate, Client Key Exchange, Certificate Verify, Change Cipher Spec, Encrypted Handshake Message
12	1.061969	LinksysG_d7:b4:b5	Netgear_93:01:08	TLS	Change Cipher Spec, Encrypted Handshake Message
13	1.310756	Netgear_93:01:08	LinksysG_d7:b4:b5	EAP	Response, EAP-TLS [RFC2716] [Aboba]
14	1.319116	LinksysG_d7:b4:b5	Netgear_93:01:08	EAP	Success
15	1.319915	LinksysG_d7:b4:b5	Netgear_93:01:08	EAPOL	Key
16	1.321178	LinksysG_d7:b4:b5	Netgear_93:01:08	EAPOL	Key
17	10.993644	Netgear_93:01:08	LinksysG_d7:b4:b5	EAPOL	Logoff
18	10.995092	LinksysG_d7:b4:b5	Netgear_93:01:08	EAP	Failure

Abbildung 5.2: Nachrichten Supplicant, Authenticator

Ab Seite 110 ist die Kommunikation zwischen Access Point und RADIUS Server abgedruckt.

	Zeit(sec)	Source	Destination		
1	0.000000	192.168.1.1	192.168.1.3	RADIUS	Access Request(1) (id=0, l=125)
2	0.001262	192.168.1.3	192.168.1.1	RADIUS	Access challenge(11) (id=0, l=99)
3	0.076907	192.168.1.1	192.168.1.3	RADIUS	Access Request(1) (id=0, l=262)
4	0.083157	192.168.1.3	192.168.1.1	RADIUS	Access challenge(11) (id=0, l=1135)
5	0.341450	192.168.1.1	192.168.1.3	RADIUS	Access Request(1) (id=0, l=158)
6	0.367402	192.168.1.3	192.168.1.1	RADIUS	Access challenge(11) (id=0, l=1127)
7	0.519855	192.168.1.1	192.168.1.3	RADIUS	Access Request(1) (id=0, l=1570)
8	0.519872	192.168.1.1	192.168.1.3	IP	Fragmented IP protocol (proto=UDP 0x11, off=1480)
9	0.523538	192.168.1.3	192.168.1.1	RADIUS	Access challenge(11) (id=0, l=99)
10	0.781107	192.168.1.1	192.168.1.3	RADIUS	Access Request(1) (id=0, l=908)
11	0.798653	192.168.1.3	192.168.1.1	RADIUS	Access challenge(11) (id=0, l=162)
12	1.054042	192.168.1.1	192.168.1.3	RADIUS	Access Request(1) (id=0, l=158)
13	1.055346	192.168.1.3	192.168.1.1	RADIUS	Access Accept(2) (id=0, l=175)

Abbildung 5.3: Nachrichten Authenticator, RADIUS Server

EAP-TLS Client, Access Point

```
Frame 1 (18 bytes on wire, 18 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.131736000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 18 bytes
    Capture Length: 18 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Start (1)
    Length: 0

0000  00 06 25 d7 b4 b5 00 09 5b 93 01 08 88 8e 01 01
0010  00 00
```

```
                                                                    # Frame 2
Frame 2 (23 bytes on wire, 23 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.132694000
    Time delta from previous packet: 0.000958000 seconds
    Time since reference or first frame: 0.000958000 seconds
    Frame Number: 2
    Packet Length: 23 bytes
    Capture Length: 23 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 5
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 0
        Length: 5
        Type: Identity [RFC2284] (1)

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 00
0010  00 05 01 00 00 05 01
```

```
                                                                    # Frame 3
Frame 3 (43 bytes on wire, 43 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.391800000
    Time delta from previous packet: 0.259106000 seconds
    Time since reference or first frame: 0.260064000 seconds
    Frame Number: 3
    Packet Length: 43 bytes
    Capture Length: 43 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
```

```
    Trailer: 00000000000000000000000000000000
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 11
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 0
        Length: 11
        Type: Identity [RFC2284] (1)
        Identity (6 bytes): thomas

0000  00 06 25 d7 b4 b5 00 09 5b 93 01 08 88 8e 01 00
0010  00 0b 02 00 00 0b 01 74 68 6f 6d 61 73 00 00 00
0020  00 00 00 00 00 00 00 00 00 00 00 00

Hinweis: Hier wird die Identität "thomas" (74 68 6f 6d 61 73)
         im Klartext übertragen.

                                                                    # Frame 4
Frame 4 (24 bytes on wire, 24 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.396280000
    Time delta from previous packet: 0.004480000 seconds
    Time since reference or first frame: 0.264544000 seconds
    Frame Number: 4
    Packet Length: 24 bytes
    Capture Length: 24 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 6
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 1
        Length: 6
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x20): Start

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 00
0010  00 06 01 01 00 06 0d 20

                                                                    # Frame 5
Frame 5 (142 bytes on wire, 142 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.468764000
    Time delta from previous packet: 0.072484000 seconds
    Time since reference or first frame: 0.337028000 seconds
    Frame Number: 5
    Packet Length: 142 bytes
    Capture Length: 142 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
    Trailer: 00000000000000000000
```

```
    Frame check sequence: 0x00000000 (incorrect, should be 0x1d643093)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 110
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 1
        Length: 110
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x80): Length
        Length: 100
        Secure Socket Layer
            TLS Record Layer: Client Hello
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 95
                Handshake Protocol: Client Hello
                    Handshake Type: Client Hello (1)
                    Length: 91
                    Version: TLS 1.0 (0x0301)
                    Random.gmt_unix_time: Mar 29, 2004 18:06:14.000000000
                    Random.bytes
                    Session ID Length: 0
                    Cipher Suites Length: 52
                    Cipher Suites (26 suites)
                        Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
                        Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
                        Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
                        Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
                        Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
                        Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
                        Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
                        Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
                        Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
                        Cipher Suite: TLS_DHE_DSS_WITH_RC4_128_SHA (0x0066)
                        Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
                        Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
                        Cipher Suite: TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x0063)
                        Cipher Suite: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x0062)
                        Cipher Suite: Unknown (0x0061)
                        Cipher Suite: TLS_DHE_RSA_WITH_DES_CBC_SHA (0x0015)
                        Cipher Suite: TLS_DHE_DSS_WITH_DES_CBC_SHA (0x0012)
                        Cipher Suite: TLS_RSA_WITH_DES_CBC_SHA (0x0009)
                        Cipher Suite: TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA (0x0065)
                        Cipher Suite: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x0064)
                        Cipher Suite: Unknown (0x0060)
                        Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA (0x0014)
                        Cipher Suite: TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA (0x0011)
                        Cipher Suite: TLS_RSA_EXPORT_WITH_DES40_CBC_SHA (0x0008)
                        Cipher Suite: TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x0006)
                        Cipher Suite: TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x0003)
                    Compression Methods Length: 1
                    Compression Methods (1 method)
                        Compression Method: null (0)

0000  00 06 25 d7 b4 b5 00 09 5b 93 01 08 88 8e 01 00
0010  00 6e 02 01 00 6e 0d 80 00 00 00 64 16 03 01 00
```

```
0020  5f 01 00 00 5b 03 01 40 68 49 76 0a ee df 95 4f
0030  10 1e b9 12 58 cb e6 03 a3 97 e1 02 b9 d3 e5 ac
0040  fe 41 3d fe 0c b9 c9 00 00 34 00 39 00 38 00 35
0050  00 16 00 13 00 0a 00 33 00 32 00 2f 00 66 00 05
0060  00 04 00 63 00 62 00 61 00 15 00 12 00 09 00 65
0070  00 64 00 60 00 14 00 11 00 08 00 06 00 03 01 00
0080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Anmerkungen:
28Byte Client.Random = 0a ee df 95 4f 10 1e b9 12 58 cb e6 03 a3 97 e1 02 b9 d3 e5
                       ac fe 41 3d fe 0c b9 c9

Ciphersuites sind die Bytepaare 00 39, 00 38, 00 35, etc.
```

Frame 6

```
Frame 6 (1052 bytes on wire, 1052 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.478999000
    Time delta from previous packet: 0.010235000 seconds
    Time since reference or first frame: 0.347263000 seconds
    Frame Number: 6
    Packet Length: 1052 bytes
    Capture Length: 1052 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 1034
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 2
        Length: 1034
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0xC0): Length More
        Length: 2040
        EAP-TLS Fragments
            Frame: 6, payload: 0-1023
            Frame: 8, payload: 1024-2039
        Secure Socket Layer
            TLS Record Layer: Server Hello
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 74
                Handshake Protocol: Server Hello
                    Handshake Type: Server Hello (2)
                    Length: 70
                    Version: TLS 1.0 (0x0301)
                    Random.gmt_unix_time: Mar 29, 2004 18:56:53.000000000
                    Random.bytes
                    Session ID Length: 32
                    Session ID (32 bytes)
                    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
                    Compression Method: null (0)
            TLS Record Layer: Certificate
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
```

```
            Length: 1802
            Handshake Protocol: Certificate
                Handshake Type: Certificate (11)
                Length: 1798
                Certificates Length: 1795
                Certificates (1795 bytes)
                    Certificate Length: 919
                    Certificate (919 bytes)
                    Certificate Length: 870
                    Certificate (870 bytes)
        TLS Record Layer: Multiple Handshake Messages
            Content Type: Handshake (22)
            Version: TLS 1.0 (0x0301)
            Length: 149
            Handshake Protocol: Certificate Request
                Handshake Type: Certificate Request (13)
                Length: 141
                Certificate types count: 2
                Certificate types (2 types)
                    Certificate type: RSA Sign (1)
                    Certificate type: DSS Sign (2)
                Distinguished Names Length: 136
                Distinguished Names (136 bytes)
                    Distinguished Name Length: 134
                    Distinguished Name (134 bytes)
            Handshake Protocol: Server Hello Done
                Handshake Type: Server Hello Done (14)
                Length: 0

Frame:
Die TLS Records Server Hello, Certificate, Certificate Request
und Server Hello Done sind fragmentiert. Frame 6 und 8 gehören
also zusammen. Für die Zusammensetzung siehe den Dump von Frame 8.

                                                                    # Frame 7
Frame 7 (38 bytes on wire, 38 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.731853000
    Time delta from previous packet: 0.252854000 seconds
    Time since reference or first frame: 0.600117000 seconds
    Frame Number: 7
    Packet Length: 38 bytes
    Capture Length: 38 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
    Trailer: 00000064160301005F0100005B03
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 6
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 2
        Length: 6
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x0):
```

```
0000  00 06 25 d7 b4 b5 00 09 5b 93 01 08 88 8e 01 00
0010  00 06 02 02 00 06 0d 00 00 00 00 64 16 03 01 00
0020  5f 01 00 00 5b 03



Frame 8 (1044 bytes on wire, 1044 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.763281000
    Time delta from previous packet: 0.031428000 seconds
    Time since reference or first frame: 0.631545000 seconds
    Frame Number: 8
    Packet Length: 1044 bytes
    Capture Length: 1044 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 1026
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 3
        Length: 1026
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x80): Length
        Length: 2040
        EAP-TLS Fragments
            Frame: 6, payload: 0-1023
            Frame: 8, payload: 1024-2039
        Secure Socket Layer
            TLS Record Layer: Server Hello
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 74
                Handshake Protocol: Server Hello
                    Handshake Type: Server Hello (2)
                    Length: 70
                    Version: TLS 1.0 (0x0301)
                    Random.gmt_unix_time: Mar 29, 2004 18:56:53.000000000
                    Random.bytes
                    Session ID Length: 32
                    Session ID (32 bytes)
                    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
                    Compression Method: null (0)
            TLS Record Layer: Certificate
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 1802
                Handshake Protocol: Certificate
                    Handshake Type: Certificate (11)
                    Length: 1798
                    Certificates Length: 1795
                    Certificates (1795 bytes)
                        Certificate Length: 919
                        Certificate (919 bytes)
                        Certificate Length: 870
```

```
                Certificate (870 bytes)
        TLS Record Layer: Multiple Handshake Messages
            Content Type: Handshake (22)
            Version: TLS 1.0 (0x0301)
            Length: 149
            Handshake Protocol: Certificate Request
                Handshake Type: Certificate Request (13)
                Length: 141
                Certificate types count: 2
                Certificate types (2 types)
                    Certificate type: RSA Sign (1)
                    Certificate type: DSS Sign (2)
                Distinguished Names Length: 136
                Distinguished Names (136 bytes)
                    Distinguished Name Length: 134
                    Distinguished Name (134 bytes)
            Handshake Protocol: Server Hello Done
                Handshake Type: Server Hello Done (14)
                Length: 0
```

Die zusammengesetzen Fragmente:

TLS Record Layer: Server Hello

```
0000  16 03 01 00 4a 02 00 00 46 03 01 40 68 55 55 aa
0010  89 7d 64 dd c1 ec c5 49 2b 1d 74 46 fc 51 d3 14
0020  be 0c de d4 fc e3 50 21 4f b0 36 20 c8 d1 86 7d
0030  80 07 c2 07 51 55 ac f5 03 05 3a c4 6a e7 c5 e9
0040  b7 1d 20 34 32 f7 5f 82 5a 55 30 eb 00 35 00
```

TLS Record Layer: Server Certificate

```
                                                  16   .. 42._.ZU0..5..
0050  03 01 07 0a 0b 00 07 06 00 07 03 00 03 97 30 82   ..............0.
0060  03 93 30 82 02 fc a0 03 02 01 02 02 01 02 30 0d   ..0...........0.
0070  06 09 2a 86 48 86 f7 0d 01 01 04 05 00 30 81 83   ..*.H........0..
0080  31 0b 30 09 06 03 55 04 06 13 02 44 45 31 16 30   1.0...U....DE1.0
0090  14 06 03 55 04 08 13 0d 4e 69 65 64 65 72 73 61   ...U....Niedersa
00a0  63 68 73 65 6e 31 15 30 13 06 03 55 04 07 13 0c   chsen1.0...U....
00b0  42 72 61 75 6e 73 63 68 77 65 69 67 31 18 30 16   Braunschweig1.0.
00c0  06 03 55 04 0a 13 0f 54 55 20 42 72 61 75 6e 73   ..U....TU Brauns
00d0  63 68 77 65 69 67 31 0b 30 09 06 03 55 04 03 13   chweig1.0...U...
00e0  02 43 41 31 1e 30 1c 06 09 2a 86 48 86 f7 0d 01   .CA1.0...*.H....
00f0  09 01 16 0f 74 2e 6f 74 74 6f 40 74 75 2d 62 73   ....t.otto@tu-bs
0100  2e 64 65 30 1e 17 0d 30 34 30 32 30 31 31 38 35   .de0...040201185
0110  37 33 39 5a 17 0d 30 36 31 30 32 38 31 38 35 37   739Z..0610281857
0120  33 39 5a 30 81 87 31 0b 30 09 06 03 55 04 06 13   39Z0..1.0...U...
0130  02 44 45 31 16 30 14 06 03 55 04 08 13 0d 4e 69   .DE1.0...U....Ni
0140  65 64 65 72 73 61 63 68 73 65 6e 31 15 30 13 06   edersachsen1.0..
0150  03 55 04 07 13 0c 42 72 61 75 6e 73 63 68 77 65   .U....Braunschwe
0160  69 67 31 18 30 16 06 03 55 04 0a 13 0f 54 55 20   ig1.0...U....TU
0170  42 72 61 75 6e 73 63 68 77 65 69 67 31 0f 30 0d   Braunschweig1.0.
0180  06 03 55 04 03 13 06 73 65 72 76 65 72 31 1e 30   ..U....server1.0
0190  1c 06 09 2a 86 48 86 f7 0d 01 09 01 16 0f 74 2e   ...*.H........t.
01a0  6f 74 74 6f 40 74 75 2d 62 73 2e 64 65 30 81 9f   otto@tu-bs.de0..
01b0  30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 03   0...*.H.........
01c0  81 8d 00 30 81 89 02 81 81 00 aa 70 cf 93 94 58   ...0.......p...X
```

```
01d0  de f5 0b 7d b6 2e 58 00 f5 16 b0 1c 19 e5 ec d6   ...}..X.........
01e0  cc 44 12 f2 c3 a4 fb 31 bb 10 67 11 65 93 af f7   .D.....1..g.e...
01f0  b8 f0 7d 40 9d 38 3f 78 43 df 94 45 7b a9 8e 97   ..}@.8?xC..E{...
0200  af 29 34 76 e4 08 23 df 6d 3c f3 36 96 21 c3 02   .)4v..#.m<.6.!..
0210  f6 5c 1d 76 5c f4 1d 5d 79 86 89 93 1b bf b9 bb   .\.v\..]y.......
0220  ad a1 f5 e0 14 54 44 2c 15 27 0a 4d 1a f2 c3 52   .....TD,.'.M...R
0230  97 ea b4 e0 8a 2e 79 94 2f 45 b1 30 de 0b 48 eb   ......y./E.0..H.
0240  29 fc db f5 79 da 5b da 8b 59 02 03 01 00 01 a3   )...y.[..Y......
0250  82 01 0f 30 82 01 0b 30 09 06 03 55 1d 13 04 02   ...0...0...U....
0260  30 00 30 2c 06 09 60 86 48 01 86 f8 42 01 0d 04   0.0,..`.H...B...
0270  1f 16 1d 4f 70 65 6e 53 53 4c 20 47 65 6e 65 72   ...OpenSSL Gener
0280  61 74 65 64 20 43 65 72 74 69 66 69 63 61 74 65   ated Certificate
0290  30 1d 06 03 55 1d 0e 04 16 04 14 e4 5e 51 53 2f   0...U.......^QS/
02a0  a3 2b d5 e2 c6 22 a2 b4 bc 0e c3 94 92 7e cd 30   .+..."........~.0
02b0  81 b0 06 03 55 1d 23 04 81 a8 30 81 a5 80 14 5f   ....U.#...0....._
02c0  70 ef f1 e5 79 cf d4 37 d2 a2 82 40 17 d3 f9 b4   p...y..7...@....
02d0  dc c9 10 a1 81 89 a4 81 86 30 81 83 31 0b 30 09   .........0..1.0.
02e0  06 03 55 04 06 13 02 44 45 31 16 30 14 06 03 55   ..U....DE1.0...U
02f0  04 08 13 0d 4e 69 65 64 65 72 73 61 63 68 73 65   ....Niedersachse
0300  6e 31 15 30 13 06 03 55 04 07 13 0c 42 72 61 75   n1.0...U....Brau
0310  6e 73 63 68 77 65 69 67 31 18 30 16 06 03 55 04   nschweig1.0...U.
0320  0a 13 0f 54 55 20 42 72 61 75 6e 73 63 68 77 65   ...TU Braunschwe
0330  69 67 31 0b 30 09 06 03 55 04 03 13 02 43 41 31   ig1.0...U....CA1
0340  1e 30 1c 06 09 2a 86 48 86 f7 0d 01 09 01 16 0f   .0...*.H........
0350  74 2e 6f 74 74 6f 40 74 75 2d 62 73 2e 64 65 82   t.otto@tu-bs.de.
0360  01 00 30 0d 06 09 2a 86 48 86 f7 0d 01 01 04 05   ..0...*.H.......
0370  00 03 81 81 00 81 23 0d e5 f8 35 7c c4 25 c2 92   ......#...5|.%..
0380  6b ba dd 78 28 64 cd f4 d0 27 3d d3 27 be 15 71   k..x(d...'=.'..q
0390  09 14 b4 04 a6 36 b8 35 b5 c5 97 6a 85 5e 2d 31   .....6.5...j.^-1
03a0  bd 17 83 3c ec 58 0b 58 90 26 e2 90 e5 3c b7 1b   ...<.X.X.&...<..
03b0  37 0e a7 c0 24 0f c2 0f 96 30 c1 72 53 47 b1 79   7...$....0.rSG.y
03c0  15 d5 91 b3 a0 fa 30 74 88 7c fd 43 0a b4 4a c8   ......0t.|.C..J.
03d0  df ed 1c 4e ea e4 c0 fe c8 70 99 d3 11 87 bf 56   ...N.....p.....V
03e0  7b 25 66 98 c4 42 90 8c c2 78 8f dc 26 e2 b5 f7   {%f..B...x..&...
03f0  a2 57 ba fd 72 00 03 66 30 82 03 62 30 82 02 cb   .W..r..f0..b0...
0400  a0 03 02 01 02 02 01 00 30 0d 06 09 2a 86 48 86   ........0...*.H.
0410  f7 0d 01 01 04 05 00 30 81 83 31 0b 30 09 06 03   .......0..1.0...
0420  55 04 06 13 02 44 45 31 16 30 14 06 03 55 04 08   U....DE1.0...U..
0430  13 0d 4e 69 65 64 65 72 73 61 63 68 73 65 6e 31   ..Niedersachsen1
0440  15 30 13 06 03 55 04 07 13 0c 42 72 61 75 6e 73   .0...U....Brauns
0450  63 68 77 65 69 67 31 18 30 16 06 03 55 04 0a 13   chweig1.0...U...
0460  0f 54 55 20 42 72 61 75 6e 73 63 68 77 65 69 67   .TU Braunschweig
0470  31 0b 30 09 06 03 55 04 03 13 02 43 41 31 1e 30   1.0...U....CA1.0
0480  1c 06 09 2a 86 48 86 f7 0d 01 09 01 16 0f 74 2e   ...*.H........t.
0490  6f 74 74 6f 40 74 75 2d 62 73 2e 64 65 30 1e 17   otto@tu-bs.de0..
04a0  0d 30 34 30 32 30 31 31 38 35 34 31 32 5a 17 0d   .040201185412Z..
04b0  31 34 30 31 32 39 31 38 35 34 31 32 5a 30 81 83   140129185412Z0..
04c0  31 0b 30 09 06 03 55 04 06 13 02 44 45 31 16 30   1.0...U....DE1.0
04d0  14 06 03 55 04 08 13 0d 4e 69 65 64 65 72 73 61   ...U....Niedersa
04e0  63 68 73 65 6e 31 15 30 13 06 03 55 04 07 13 0c   chsen1.0...U....
04f0  42 72 61 75 6e 73 63 68 77 65 69 67 31 18 30 16   Braunschweig1.0.
0500  06 03 55 04 0a 13 0f 54 55 20 42 72 61 75 6e 73   ..U....TU Brauns
0510  63 68 77 65 69 67 31 0b 30 09 06 03 55 04 03 13   chweig1.0...U...
0520  02 43 41 31 1e 30 1c 06 09 2a 86 48 86 f7 0d 01   .CA1.0...*.H....
0530  09 01 16 0f 74 2e 6f 74 74 6f 40 74 75 2d 62 73   ....t.otto@tu-bs
0540  2e 64 65 30 81 9f 30 0d 06 09 2a 86 48 86 f7 0d   .de0..0...*.H...
0550  01 01 01 05 00 03 81 8d 00 30 81 89 02 81 81 00   .........0......
0560  ec ac 74 45 4a 96 a0 91 cb 66 77 a1 49 b3 e0 8c   ..tEJ....fw.I...
```

```
0570  94 b7 bf ba 13 df 7a a3 15 54 3a be 76 8d 1f 64   ......z..T:.v..d
0580  aa 93 a4 1f 65 01 de e8 b8 5b fb 1a b4 78 2a d1   ....e....[...x*.
0590  2c a1 71 b7 62 63 66 32 e4 cf 57 01 40 58 1e 7e   ,.q.bcf2..W.@X.~
05a0  c1 b9 35 19 94 f1 e2 f7 f7 b0 29 72 76 26 4f 37   ..5.......)rv&O7
05b0  39 4b 95 34 f1 57 07 e7 88 31 69 c0 d6 1a 02 72   9K.4.W...1i....r
05c0  d9 c5 af 4a c9 8e af b8 99 df 42 d4 9d 33 ac f7   ...J......B..3..
05d0  e1 4f c3 75 53 ff 01 c7 1a f9 19 61 82 f8 eb 53   .O.uS......a...S
05e0  02 03 01 00 01 a3 81 e3 30 81 e0 30 1d 06 03 55   ........0..0...U
05f0  1d 0e 04 16 04 14 5f 70 ef f1 e5 79 cf d4 37 d2   ......_p...y..7.
0600  a2 82 40 17 d3 f9 b4 dc c9 10 30 81 b0 06 03 55   ..@.......0....U
0610  1d 23 04 81 a8 30 81 a5 80 14 5f 70 ef f1 e5 79   .#...0...._p...y
0620  cf d4 37 d2 a2 82 40 17 d3 f9 b4 dc c9 10 a1 81   ..7...@.........
0630  89 a4 81 86 30 81 83 31 0b 30 09 06 03 55 04 06   ....0..1.0...U..
0640  13 02 44 45 31 16 30 14 06 03 55 04 08 13 0d 4e   ..DE1.0...U....N
0650  69 65 64 65 72 73 61 63 68 73 65 6e 31 15 30 13   iedersachsen1.0.
0660  06 03 55 04 07 13 0c 42 72 61 75 6e 73 63 68 77   ..U....Braunschw
0670  65 69 67 31 18 30 16 06 03 55 04 0a 13 0f 54 55   eig1.0...U....TU
0680  20 42 72 61 75 6e 73 63 68 77 65 69 67 31 0b 30    Braunschweig1.0
0690  09 06 03 55 04 03 13 02 43 41 31 1e 30 1c 06 09   ...U....CA1.0...
06a0  2a 86 48 86 f7 0d 01 09 01 16 0f 74 2e 6f 74 74   *.H........t.ott
06b0  6f 40 74 75 2d 62 73 2e 64 65 82 01 00 30 0c 06   o@tu-bs.de...0..
06c0  03 55 1d 13 04 05 30 03 01 01 ff 30 0d 06 09 2a   .U....0....0...*
06d0  86 48 86 f7 0d 01 01 04 05 00 03 81 81 00 b8 58   .H.............X
06e0  b6 85 8c e0 81 4d fa 3d 69 c1 3f 9e ab f4 5d c0   .....M.=i.?...].
06f0  30 0e 1e dc 38 25 f5 80 4f c0 4e a3 2b b2 bd 6e   0...8%..O.N.+..n
0700  db 37 64 28 2f 7e 80 eb 08 92 c1 2e 7f a5 06 20   .7d(/~.........
0710  c0 1e 7a 65 c8 98 54 73 91 12 da 88 27 18 4b e4   ..ze..Ts....'.K.
0720  6d 5d 0c 5e 70 df cf db 36 d4 46 5d 63 ef d8 34   m].^p...6.F]c..4
0730  5c 23 7f f4 30 0f 1a e7 f8 f5 27 ba ab 4d c0 bc   \#..0.....'..M..
0740  1d e2 bc ed ee b8 90 ad f8 5c 0e 6a 06 6d 1a 8c   .........\.j.m..
0750  14 75 56 2b 9b 4c a4 b8 0b d8 b9 30 e2 ff

TLS Record Layer: Multiple Handshake Messages

                                                  16 03   .uV+.L.....0....
0760  01 00 95

Handshake Protocol: Certificate Request

              0d 00 00 8d 02 01 02 00 88 00 86 30 81   ..............0.
0770  83 31 0b 30 09 06 03 55 04 06 13 02 44 45 31 16   .1.0...U....DE1.
0780  30 14 06 03 55 04 08 13 0d 4e 69 65 64 65 72 73   0...U....Nieders
0790  61 63 68 73 65 6e 31 15 30 13 06 03 55 04 07 13   achsen1.0...U...
07a0  0c 42 72 61 75 6e 73 63 68 77 65 69 67 31 18 30   .Braunschweig1.0
07b0  16 06 03 55 04 0a 13 0f 54 55 20 42 72 61 75 6e   ...U....TU Braun
07c0  73 63 68 77 65 69 67 31 0b 30 09 06 03 55 04 03   schweig1.0...U..
07d0  13 02 43 41 31 1e 30 1c 06 09 2a 86 48 86 f7 0d   ..CA1.0...*.H...
07e0  01 09 01 16 0f 74 2e 6f 74 74 6f 40 74 75 2d 62   .....t.otto@tu-b
07f0  73 2e 64 65                                      s.de....

Handshake Protocol: Server Hello Done

                 0e 00 00 00
```

Frame 9

```
Frame 9 (1440 bytes on wire, 1440 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.909495000
    Time delta from previous packet: 0.146214000 seconds
```

```
    Time since reference or first frame: 0.777759000 seconds
    Frame Number: 9
    Packet Length: 1440 bytes
    Capture Length: 1440 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
    Trailer: 00000000000000000000
    Frame check sequence: 0x00000000 (incorrect, should be 0x80b15ea7)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 1408
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 3
        Length: 1408
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0xC0): Length More
        Length: 2144
        EAP-TLS Fragments
            Frame: 9, payload: 0-1397
            Frame: 11, payload: 1398-2143
        Secure Socket Layer
            TLS Record Layer: Certificate
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 1802
                Handshake Protocol: Certificate
                    Handshake Type: Certificate (11)
                    Length: 1798
                    Certificates Length: 1795
                    Certificates (1795 bytes)
                        Certificate Length: 919
                        Certificate (919 bytes)
                        Certificate Length: 870
                        Certificate (870 bytes)
            TLS Record Layer: Client Key Exchange
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 134
                Handshake Protocol: Client Key Exchange
                    Handshake Type: Client Key Exchange (16)
                    Length: 130
            TLS Record Layer: Certificate Verify
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 134
                Handshake Protocol: Certificate Verify
                    Handshake Type: Certificate Verify (15)
                    Length: 130
            TLS Record Layer: Change Cipher Spec
                Content Type: Change Cipher Spec (20)
                Version: TLS 1.0 (0x0301)
                Length: 1
                Change Cipher Spec Message
            TLS Record Layer: Encrypted Handshake Message
```

```
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 48
                Handshake Protocol: Encrypted Handshake Message

Frame:
Auch in Richtung Client->Server muss fragmentiert werden. Die Stelle,
an der der Payload von Frame 11 ansetzt, ist im folgenden kenntlich gemacht.

TLS Record Layer: Certificate

0000  16 03 01 07 0a 0b 00 07 06 00 07 03 00 03 97 30   ...............0
0010  82 03 93 30 82 02 fc a0 03 02 01 02 02 01 01 30   ...0...........0
0020  0d 06 09 2a 86 48 86 f7 0d 01 01 04 05 00 30 81   ...*.H........0.
0030  83 31 0b 30 09 06 03 55 04 06 13 02 44 45 31 16   .1.0...U....DE1.
0040  30 14 06 03 55 04 08 13 0d 4e 69 65 64 65 72 73   0...U....Nieders
0050  61 63 68 73 65 6e 31 15 30 13 06 03 55 04 07 13   achsen1.0...U...
0060  0c 42 72 61 75 6e 73 63 68 77 65 69 67 31 18 30   .Braunschweig1.0
0070  16 06 03 55 04 0a 13 0f 54 55 20 42 72 61 75 6e   ...U....TU Braun
0080  73 63 68 77 65 69 67 31 0b 30 09 06 03 55 04 03   schweig1.0...U..
0090  13 02 43 41 31 1e 30 1c 06 09 2a 86 48 86 f7 0d   ..CA1.0...*.H...
00a0  01 09 01 16 0f 74 2e 6f 74 74 6f 40 74 75 2d 62   .....t.otto@tu-b
00b0  73 2e 64 65 30 1e 17 0d 30 34 30 32 30 31 31 38   s.de0...04020118
00c0  35 36 35 36 5a 17 0d 30 36 31 30 32 38 31 38 35   5656Z..061028185
00d0  36 35 36 5a 30 81 87 31 0b 30 09 06 03 55 04 06   656Z0..1.0...U..
00e0  13 02 44 45 31 16 30 14 06 03 55 04 08 13 0d 4e   ..DE1.0...U....N
00f0  69 65 64 65 72 73 61 63 68 73 65 6e 31 15 30 13   iedersachsen1.0.
0100  06 03 55 04 07 13 0c 42 72 61 75 6e 73 63 68 77   ..U....Braunschw
0110  65 69 67 31 18 30 16 06 03 55 04 0a 13 0f 54 55   eig1.0...U....TU
0120  20 42 72 61 75 6e 73 63 68 77 65 69 67 31 0f 30    Braunschweig1.0
0130  0d 06 03 55 04 03 13 06 74 68 6f 6d 61 73 31 1e   ...U....thomas1.
0140  30 1c 06 09 2a 86 48 86 f7 0d 01 09 01 16 0f 74   0...*.H........t
0150  2e 6f 74 74 6f 40 74 75 2d 62 73 2e 64 65 30 81   .otto@tu-bs.de0.
0160  9f 30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00   .0...*.H........
0170  03 81 8d 00 30 81 89 02 81 81 00 c4 c0 7e e6 c7   ....0........~..
0180  ba 5d b6 97 44 0f 86 bd 3d 9d d4 bb ec c5 c6 98   .]..D...=.......
0190  d1 64 fa 00 f6 d7 23 67 3e 3e bc f0 de f8 17 1e   .d....#g>>......
01a0  77 dc ed e0 a7 1b 0d 9e c4 c8 6d c5 0b f1 62 5b   w.........m...b[
01b0  14 89 bd 55 cf 8e f3 2b 14 b9 a3 dc 68 45 86 73   ...U...+....hE.s
01c0  88 58 10 b9 b8 fa 14 3e d9 f2 ae 45 4f 30 4e 2c   .X.....>...EO0N,
01d0  a1 02 a0 17 6a d3 2e 87 ec a9 75 a3 7e d5 a7 1f   ....j.....u.~...
01e0  e0 a7 9f ab 1e 08 b8 a1 31 22 1e 63 10 10 29 69   ........1".c..)i
01f0  91 9d 0d ff 2c 64 44 82 1d 8e ff 02 03 01 00 01   ....,dD.........
0200  a3 82 01 0f 30 82 01 0b 30 09 06 03 55 1d 13 04   ....0...0...U...
0210  02 30 00 30 2c 06 09 60 86 48 01 86 f8 42 01 0d   .0.0,..‘.H...B..
0220  04 1f 16 1d 4f 70 65 6e 53 53 4c 20 47 65 6e 65   ....OpenSSL Gene
0230  72 61 74 65 64 20 43 65 72 74 69 66 69 63 61 74   rated Certificat
0240  65 30 1d 06 03 55 1d 0e 04 16 04 14 59 ad f1 df   e0...U......Y...
0250  fb d7 ec e6 89 5f 44 66 90 44 fc 5b 7c 54 79 00   ....._Df.D.[|Ty.
0260  30 81 b0 06 03 55 1d 23 04 81 a8 30 81 a5 80 14   0....U.#...0....
0270  5f 70 ef f1 e5 79 cf d4 37 d2 a2 82 40 17 d3 f9   _p...y..7...@...
0280  b4 dc c9 10 a1 81 89 a4 81 86 30 81 83 31 0b 30   ..........0..1.0
0290  09 06 03 55 04 06 13 02 44 45 31 16 30 14 06 03   ...U....DE1.0...
02a0  55 04 08 13 0d 4e 69 65 64 65 72 73 61 63 68 73   U....Niedersachs
02b0  65 6e 31 15 30 13 06 03 55 04 07 13 0c 42 72 61   en1.0...U....Bra
02c0  75 6e 73 63 68 77 65 69 67 31 18 30 16 06 03 55   unschweig1.0...U
02d0  04 0a 13 0f 54 55 20 42 72 61 75 6e 73 63 68 77   ....TU Braunschw
02e0  65 69 67 31 0b 30 09 06 03 55 04 03 13 02 43 41   eig1.0...U....CA
```

```
02f0  31 1e 30 1c 06 09 2a 86 48 86 f7 0d 01 09 01 16   1.0...*.H.......
0300  0f 74 2e 6f 74 74 6f 40 74 75 2d 62 73 2e 64 65   .t.otto@tu-bs.de
0310  82 01 00 30 0d 06 09 2a 86 48 86 f7 0d 01 01 04   ...0...*.H......
0320  05 00 03 81 81 00 44 48 9e 14 31 56 4d 3b f9 8a   ......DH..1VM;..
0330  dd 50 fc d7 25 a9 26 a8 5e 46 8e 82 0a 66 cf 92   .P..%.&.^F...f..
0340  f1 b8 20 59 0f 15 26 47 aa 12 83 cf ab 7d 46 7c   .. Y..&G.....}F|
0350  7a 68 b1 f1 73 60 4e 97 53 39 de f2 dd 50 00 c1   zh..s`N.S9...P..
0360  82 e1 22 82 97 ba 8f f4 84 df 1f c3 ad d8 f3 4d   ..".............M
0370  fd da c0 40 f1 29 ce 23 98 a7 32 b5 eb 29 5d 89   ...@.).#..2..)].
0380  cb f3 1a 31 b5 fc 58 58 d0 9d 82 31 6b ef 82 f1   ...1..XX...1k...
0390  c3 32 7f 2e db f7 0b bc 00 bf fb ae 37 47 ba 32   .2..........7G.2
03a0  fe 48 21 51 59 0e 00 03 66 30 82 03 62 30 82 02   .H!QY...f0..b0..
03b0  cb a0 03 02 01 02 02 01 00 30 0d 06 09 2a 86 48   .........0...*.H
03c0  86 f7 0d 01 01 04 05 00 30 81 83 31 0b 30 09 06   ........0..1.0..
03d0  03 55 04 06 13 02 44 45 31 16 30 14 06 03 55 04   .U....DE1.0...U.
03e0  08 13 0d 4e 69 65 64 65 72 73 61 63 68 73 65 6e   ...Niedersachsen
03f0  31 15 30 13 06 03 55 04 07 13 0c 42 72 61 75 6e   1.0...U....Braun
0400  73 63 68 77 65 69 67 31 18 30 16 06 03 55 04 0a   schweig1.0...U..
0410  13 0f 54 55 20 42 72 61 75 6e 73 63 68 77 65 69   ..TU Braunschwei
0420  67 31 0b 30 09 06 03 55 04 03 13 02 43 41 31 1e   g1.0...U....CA1.
0430  30 1c 06 09 2a 86 48 86 f7 0d 01 09 01 16 0f 74   0...*.H........t
0440  2e 6f 74 74 6f 40 74 75 2d 62 73 2e 64 65 30 1e   .otto@tu-bs.de0.
0450  17 0d 30 34 30 32 30 31 31 38 35 34 31 32 5a 17   ..040201185412Z.
0460  0d 31 34 30 31 32 39 31 38 35 34 31 32 5a 30 81   .140129185412Z0.
0470  83 31 0b 30 09 06 03 55 04 06 13 02 44 45 31 16   .1.0...U....DE1.
0480  30 14 06 03 55 04 08 13 0d 4e 69 65 64 65 72 73   0...U....Nieders
0490  61 63 68 73 65 6e 31 15 30 13 06 03 55 04 07 13   achsen1.0...U...
04a0  0c 42 72 61 75 6e 73 63 68 77 65 69 67 31 18 30   .Braunschweig1.0
04b0  16 06 03 55 04 0a 13 0f 54 55 20 42 72 61 75 6e   ...U....TU Braun
04c0  73 63 68 77 65 69 67 31 0b 30 09 06 03 55 04 03   schweig1.0...U..
04d0  13 02 43 41 31 1e 30 1c 06 09 2a 86 48 86 f7 0d   ..CA1.0...*.H...
04e0  01 09 01 16 0f 74 2e 6f 74 74 6f 40 74 75 2d 62   .....t.otto@tu-b
04f0  73 2e 64 65 30 81 9f 30 0d 06 09 2a 86 48 86 f7   s.de0..0...*.H..
0500  0d 01 01 01 05 00 03 81 8d 00 30 81 89 02 81 81   ..........0.....
0510  00 ec ac 74 45 4a 96 a0 91 cb 66 77 a1 49 b3 e0   ...tEJ....fw.I..
0520  8c 94 b7 bf ba 13 df 7a a3 15 54 3a be 76 8d 1f   .......z..T:.v..
0530  64 aa 93 a4 1f 65 01 de e8 b8 5b fb 1a b4 78 2a   d....e....[...x*
0540  d1 2c a1 71 b7 62 63 66 32 e4 cf 57 01 40 58 1e   .,.q.bcf2..W.@X.
0550  7e c1 b9 35 19 94 f1 e2 f7 f7 b0 29 72 76 26 4f   ~..5.......)rv&O
0560  37 39 4b 95 34 f1 57 07 e7 88 31 69 c0 d6 1a 02   79K.4.W...1i....
0570  72 d9 c5 af 4a c9                                 r...J.
```

Fragment: Frame 11 Payload

```
                      8e af b8 99 df 42 d4 9d 33 ac   r...J......B..3.
0580  f7 e1 4f c3 75 53 ff 01 c7 1a f9 19 61 82 f8 eb   ..O.uS......a...
0590  53 02 03 01 00 01 a3 81 e3 30 81 e0 30 1d 06 03   S........0..0...
05a0  55 1d 0e 04 16 04 14 5f 70 ef f1 e5 79 cf d4 37   U......_p...y..7
05b0  d2 a2 82 40 17 d3 f9 b4 dc c9 10 30 81 b0 06 03   ...@.......0....
05c0  55 1d 23 04 81 a8 30 81 a5 80 14 5f 70 ef f1 e5   U.#...0....._p...
05d0  79 cf d4 37 d2 a2 82 40 17 d3 f9 b4 dc c9 10 a1   y..7...@........
05e0  81 89 a4 81 86 30 81 83 31 0b 30 09 06 03 55 04   .....0..1.0...U.
05f0  06 13 02 44 45 31 16 30 14 06 03 55 04 08 13 0d   ...DE1.0...U....
0600  4e 69 65 64 65 72 73 61 63 68 73 65 6e 31 15 30   Niedersachsen1.0
0610  13 06 03 55 04 07 13 0c 42 72 61 75 6e 73 63 68   ...U....Braunsch
0620  77 65 69 67 31 18 30 16 06 03 55 04 0a 13 0f 54   weig1.0...U....T
0630  55 20 42 72 61 75 6e 73 63 68 77 65 69 67 31 0b   U Braunschweig1.
0640  30 09 06 03 55 04 03 13 02 43 41 31 1e 30 1c 06   0...U....CA1.0..
```

```
0650  09 2a 86 48 86 f7 0d 01 09 01 16 0f 74 2e 6f 74   .*.H........t.ot
0660  74 6f 40 74 75 2d 62 73 2e 64 65 82 01 00 30 0c   to@tu-bs.de...0.
0670  06 03 55 1d 13 04 05 30 03 01 01 ff 30 0d 06 09   ..U....0....0...
0680  2a 86 48 86 f7 0d 01 01 04 05 00 03 81 81 00 b8   *.H.............
0690  58 b6 85 8c e0 81 4d fa 3d 69 c1 3f 9e ab f4 5d   X.....M.=i.?...]
06a0  c0 30 0e 1e dc 38 25 f5 80 4f c0 4e a3 2b b2 bd   .0...8%..O.N.+..
06b0  6e db 37 64 28 2f 7e 80 eb 08 92 c1 2e 7f a5 06   n.7d(/~.........
06c0  20 c0 1e 7a 65 c8 98 54 73 91 12 da 88 27 18 4b    ..ze..Ts....'.K
06d0  e4 6d 5d 0c 5e 70 df cf db 36 d4 46 5d 63 ef d8   .m].^p...6.F]c..
06e0  34 5c 23 7f f4 30 0f 1a e7 f8 f5 27 ba ab 4d c0   4\#..0.....'..M.
06f0  bc 1d e2 bc ed ee b8 90 ad f8 5c 0e 6a 06 6d 1a   ..........\.j.m.
0700  8c 14 75 56 2b 9b 4c a4 b8 0b d8 b9 30 e2 ff

TLS Record Layer: Client Key Exchange
                                                  16   ..uV+.L.....0...
0710  03 01 00 86 10 00 00 82 00 80 93 d9 34 31 5e 61   ............41^a
0720  72 3f 13 61 30 41 69 a5 a4 af 1b 5a 3b 85 9d b6   r?.a0Ai....Z;...
0730  2b c2 eb b1 c7 5a db e6 72 b2 65 bc 8b 09 01 03   +....Z..r.e.....
0740  33 df 9d da 6b 0a 40 31 04 d5 79 6e 8d 8e 71 02   3...k.@1..yn..q.
0750  5b 92 83 d9 f7 0f c5 5d 74 55 8d 77 cf 3b 1f 8c   [.......]tU.w.;..
0760  56 0c f4 e6 58 be e7 c5 ff f2 77 18 d9 55 c7 57   V...X.....w..U.W
0770  96 67 70 23 8e f0 27 28 9b 1e 1e 49 62 af 73 d7   .gp#..'(...Ib.s.
0780  a7 7a 57 39 85 f3 39 2b 0e 55 ea c6 3b bd 7a b5   .zW9..9+.U..;.z.
0790  0c 28 4a c4 44 66 1b 51 f3 04

TLS Record Layer: Certificate Verify

                                     16 03 01 00 86 0f   .(J.Df.Q........
07a0  00 00 82 00 80 2f df 00 53 f1 4d 4e de d0 ba 34   ...../..S.MN...4
07b0  bf ee 41 56 87 e0 02 ac dc 0e 62 ac 0a 58 bb 58   ..AV......b..X.X
07c0  38 dc f5 be ca e3 a5 00 54 da 6e e9 e6 02 ee 9b   8.......T.n.....
07d0  e4 6e 8f 76 cb 92 ac 08 63 99 af c5 83 9e 1c e0   .n.v....c.......
07e0  14 91 4b 02 d6 00 c2 bf 43 42 3c 75 fe 20 fa 82   ..K.....CB<u. ..
07f0  16 08 d0 30 42 90 55 2d 0c c6 3f 47 b8 50 42 0f   ...0B.U-..?G.PB.
0800  0c 59 96 2c 1b 9d 09 9e bd 3f 32 ec ca fd 49 5a   .Y.,.....?2...IZ
0810  18 a1 48 05 fe 89 f1 74 77 7e 33 96 05 96 e3 9c   ..H....tw~3.....
0820  ac 59 5c b4 62

TLS Record Layer: Change Cipher Spec

                    14 03 01 00 01 01

TLS Record Layer: Encrypted Handshake Message

                      16 03 01 00 30   .Y\.b.........0
0830  8e 4f 7f 60 ab 18 ee 59 0f 0a 91 25 f6 43 dc b9   .O.`...Y...%.C..
0840  3c 6f af 8b d9 a2 82 62 0d d6 a7 44 08 94 e7 95   <o.....b...D....
0850  bf a5 40 7b ad ee ab b9 3c dd 2a a0 2e 22 a7 1a   ..@{....<.*.."..
```

Frame 10

```
Frame 10 (24 bytes on wire, 24 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:14.918563000
    Time delta from previous packet: 0.009068000 seconds
    Time since reference or first frame: 0.786827000 seconds
    Frame Number: 10
    Packet Length: 24 bytes
    Capture Length: 24 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
```

```
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 6
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 4
        Length: 6
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x0):

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 00   ..[.....%.......
0010  00 06 01 04 00 06 0d 00                           ........

                                                                    # Frame 11
Frame 11 (784 bytes on wire, 784 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:15.171831000
    Time delta from previous packet: 0.253268000 seconds
    Time since reference or first frame: 1.040095000 seconds
    Frame Number: 11
    Packet Length: 784 bytes
    Capture Length: 784 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
    Trailer: 09060355040313024341
    Frame check sequence: 0x311e301c (incorrect, should be 0x1a5bddf7)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 752
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 4
        Length: 752
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x0):
        EAP-TLS Fragments
            Frame: 9, payload: 0-1397
            Frame: 11, payload: 1398-2143
        Secure Socket Layer
            TLS Record Layer: Certificate
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 1802
                Handshake Protocol: Certificate
                    Handshake Type: Certificate (11)
                    Length: 1798
                    Certificates Length: 1795
                    Certificates (1795 bytes)
                        Certificate Length: 919
                        Certificate (919 bytes)
                        Certificate Length: 870
                        Certificate (870 bytes)
```

```
            TLS Record Layer: Client Key Exchange
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 134
                Handshake Protocol: Client Key Exchange
                    Handshake Type: Client Key Exchange (16)
                    Length: 130
            TLS Record Layer: Certificate Verify
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 134
                Handshake Protocol: Certificate Verify
                    Handshake Type: Certificate Verify (15)
                    Length: 130
            TLS Record Layer: Change Cipher Spec
                Content Type: Change Cipher Spec (20)
                Version: TLS 1.0 (0x0301)
                Length: 1
                Change Cipher Spec Message
            TLS Record Layer: Encrypted Handshake Message
                Content Type: Handshake (22)
                Version: TLS 1.0 (0x0301)
                Length: 48
                Handshake Protocol: Encrypted Handshake Message

Frame: s. Frame 9

                                                                    # Frame 12
Frame 12 (87 bytes on wire, 87 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:15.193705000
    Time delta from previous packet: 0.021874000 seconds
    Time since reference or first frame: 1.061969000 seconds
    Frame Number: 12
    Packet Length: 87 bytes
    Capture Length: 87 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 69
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 5
        Length: 69
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x80): Length
        Length: 59
        Secure Socket Layer
            TLS Record Layer: Change Cipher Spec
                Content Type: Change Cipher Spec (20)
                Version: TLS 1.0 (0x0301)
                Length: 1
                Change Cipher Spec Message
            TLS Record Layer: Encrypted Handshake Message
                Content Type: Handshake (22)
```

```
                Version: TLS 1.0 (0x0301)
                Length: 48
                Handshake Protocol: Encrypted Handshake Message

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 00
0010  00 45 01 05 00 45 0d 80 00 00 00 3b

TLS Record Layer: Change Cipher Spec

                                          14 03 01 00
0020  01 01

TLS Record Layer: Encrypted Handshake Message

            16 03 01 00 30 96 ea fb 19 05 ad 0c 82 bc
0030  f8 db 37 6c c8 64 56 9e 4f fe 01 fe d4 71 17 75
0040  33 de dd 5b 06 fa 3f 18 86 bc 74 75 e2 fe 88 af
0050  13 57 06 2b 1c b7 a0

                                                                    # Frame 13
Frame 13 (38 bytes on wire, 38 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:15.442492000
    Time delta from previous packet: 0.248787000 seconds
    Time since reference or first frame: 1.310756000 seconds
    Frame Number: 13
    Packet Length: 38 bytes
    Capture Length: 38 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
    Trailer: 8EAFB899DF42D49D33ACF7E14FC3
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 6
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 5
        Length: 6
        Type: EAP-TLS [RFC2716] [Aboba] (13)
        Flags(0x0):

0000  00 06 25 d7 b4 b5 00 09 5b 93 01 08 88 8e 01 00
0010  00 06 02 05 00 06 0d 00 8e af b8 99 df 42 d4 9d
0020  33 ac f7 e1 4f c3

                                                                    # Frame 14
Frame 14 (22 bytes on wire, 22 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:15.450852000
    Time delta from previous packet: 0.008360000 seconds
    Time since reference or first frame: 1.319116000 seconds
    Frame Number: 14
    Packet Length: 22 bytes
    Capture Length: 22 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
```

```
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 4
    Extensible Authentication Protocol
        Code: Success (3)
        Id: 5
        Length: 4
                                      Version 1
                                  802.1x    EAP-Packet
                                           |----|--|--|
0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 00
0010  00 04 03 05 00 04
      |---|
      Length
```

Es folgt die Zustellung über den RC4-Keydescriptor, in Frame 15
der Broadcast-Schlüssel, in Frame 16 der Unicast-Schlüssel.

```
                                                                    # Frame 15
Frame 15 (75 bytes on wire, 75 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:15.451651000
    Time delta from previous packet: 0.000799000 seconds
    Time since reference or first frame: 1.319915000 seconds
    Frame Number: 15
    Packet Length: 75 bytes
    Capture Length: 75 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Key (3)
    Length: 57
    Descriptor Type: RC4 Descriptor (1)
    Key Length: 13
    Replay Counter: 9487538100469077781
    Key IV: 8C9E70CD4F00100000000070A4BB2AC0
    Key Index: broadcast, index 0
    0... .... = Key Type: Broadcast
    .000 0000 = Index Number: 0
    Key Signature: 014B1490A55749572CE9BD77916404B9
    Key: 49CEFD63734D996027BBE4271D

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 03
0010  00 39 01 00 0d 83 aa 81 a2 9f 71 97 15 8c 9e 70
0020  cd 4f 00 10 00 00 00 00 70 a4 bb 2a c0 00 01 4b
0030  14 90 a5 57 49 57 2c e9 bd 77 91 64 04 b9 49 ce
0040  fd 63 73 4d 99 60 27 bb e4 27 1d

                                                                    # Frame 16
Frame 16 (62 bytes on wire, 62 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:15.452914000
    Time delta from previous packet: 0.001263000 seconds
    Time since reference or first frame: 1.321178000 seconds
    Frame Number: 16
```

```
    Packet Length: 62 bytes
    Capture Length: 62 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Key (3)
    Length: 44
    Descriptor Type: RC4 Descriptor (1)
    Key Length: 13
    Replay Counter: 9487538100475661966
    Key IV: E59F3A2300000000000000000000000000
    Key Index: unicast, index 3
    1... .... = Key Type: Unicast
    .000 0011 = Index Number: 3
    Key Signature: 977D9A56F0C07CF056D9F80F6459DFF6
[Malformed Packet: EAPOL]

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 03
0010  00 2c 01 00 0d 83 aa 81 a2 9f d6 0e 8e e5 9f 3a
0020  23 00 00 00 00 00 00 00 00 00 00 00 00 83 97 7d
0030  9a 56 f0 c0 7c f0 56 d9 f8 0f 64 59 df f6

                                                                # Frame 17
Frame 17 (18 bytes on wire, 18 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:25.125380000
    Time delta from previous packet: 9.672466000 seconds
    Time since reference or first frame: 10.993644000 seconds
    Frame Number: 17
    Packet Length: 18 bytes
    Capture Length: 18 bytes
Ethernet II, Src: 00:09:5b:93:01:08, Dst: 00:06:25:d7:b4:b5
    Destination: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Source: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Logoff (2)
    Length: 0

0000  00 06 25 d7 b4 b5 00 09 5b 93 01 08 88 8e 01 02
0010  00 00

                                                                # Frame 18
Frame 18 (22 bytes on wire, 22 bytes captured)
    Arrival Time: Mar 29, 2004 18:06:25.126828000
    Time delta from previous packet: 0.001448000 seconds
    Time since reference or first frame: 10.995092000 seconds
    Frame Number: 18
    Packet Length: 22 bytes
    Capture Length: 22 bytes
Ethernet II, Src: 00:06:25:d7:b4:b5, Dst: 00:09:5b:93:01:08
    Destination: 00:09:5b:93:01:08 (Netgear_93:01:08)
    Source: 00:06:25:d7:b4:b5 (LinksysG_d7:b4:b5)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
```

```
    Version: 1
    Type: EAP Packet (0)
    Length: 4
    Extensible Authentication Protocol
        Code: Failure (4)
        Id: 6
        Length: 4

0000  00 09 5b 93 01 08 00 06 25 d7 b4 b5 88 8e 01 00
0010  00 04 04 06 00 04
```

EAP-TLS Access Point, RADIUS Server

```
Frame 1 (167 bytes on wire, 167 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:53.740229000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 167 bytes
    Capture Length: 167 bytes
Ethernet II, Src: 00:06:25:d7:b4:b3, Dst: 00:e0:18:89:2d:16
    Destination: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Source: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.1 (192.168.1.1),
                  Dst Addr: 192.168.1.3 (192.168.1.3)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 153
    Identification: 0xb4b9 (46265)
    Flags: 0x04
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (0x11)
    Header checksum: 0x0246 (correct)
    Source: 192.168.1.1 (192.168.1.1)
    Destination: 192.168.1.3 (192.168.1.3)
User Datagram Protocol, Src Port: nfs (2049), Dst Port: radius (1812)
    Source port: nfs (2049)
    Destination port: radius (1812)
    Length: 133
    Checksum: 0x7e60 (correct)
Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x0 (0)
    Length: 125
    Authenticator: 0x2CD95F870000000000000000000000000
    Attribute value pairs
        t:User Name(1) l:8, Value:"thomas"
        t:NAS IP Address(4) l:6, Value:192.168.1.1
        t:Called Station Id(30) l:14, Value:"000625d7b4b5"
        t:Calling Station Id(31) l:14, Value:"00095b930108"
        t:NAS identifier(32) l:14, Value:"000625d7b4b5"
        t:NAS Port(5) l:6, Value:26
        t:Framed MTU(12) l:6, Value:1400
        t:NAS Port Type(61) l:6, Value:Wireless IEEE 802.11(19)
        t:EAP Message(79) l:13
            Extensible Authentication Protocol
                Code: Response (2)
                Id: 0
                Length: 11
                Type: Identity [RFC2284] (1)
```

```
               Identity (6 bytes): thomas
        t:Message Authenticator(80) l:18, Value:4F39A19A45A2AB9CFEA9B1BB6638D19E

0000  00 e0 18 89 2d 16 00 06 25 d7 b4 b3 08 00 45 00   ....-...%.....E.
0010  00 99 b4 b9 40 00 40 11 02 46 c0 a8 01 01 c0 a8   ....@.@..F......
0020  01 03 08 01 07 14 00 85 7e 60 01 00 00 7d 2c d9   ........~`...},.
0030  5f 87 00 00 00 00 00 00 00 00 00 00 00 01 08      _..............
0040  74 68 6f 6d 61 73 04 06 c0 a8 01 01 1e 0e 30 30   thomas........00
0050  30 36 32 35 64 37 62 34 62 35 1f 0e 30 30 30 39   0625d7b4b5..0009
0060  35 62 39 33 30 31 30 38 20 0e 30 30 30 36 32 35   5b930108 .000625
0070  64 37 62 34 62 35 05 06 00 00 00 1a 0c 06 00 00   d7b4b5..........
0080  05 78 3d 06 00 00 00 13 4f 0d 02 00 00 0b 01 74   .x=.....O......t
0090  68 6f 6d 61 73 50 12 4f 39 a1 9a 45 a2 ab 9c fe   homasP.O9..E....
00a0  a9 b1 bb 66 38 d1 9e                              ...f8..
```

Frame 2

```
Frame 2 (141 bytes on wire, 141 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:53.741491000
    Time delta from previous packet: 0.001262000 seconds
    Time since reference or first frame: 0.001262000 seconds
    Frame Number: 2
    Packet Length: 141 bytes
    Capture Length: 141 bytes
Ethernet II, Src: 00:e0:18:89:2d:16, Dst: 00:06:25:d7:b4:b3
    Destination: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Source: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.3 (192.168.1.3),
                  Dst Addr: 192.168.1.1 (192.168.1.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 127
    Identification: 0x0018 (24)
    Flags: 0x04
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (0x11)
    Header checksum: 0xb701 (correct)
    Source: 192.168.1.3 (192.168.1.3)
    Destination: 192.168.1.1 (192.168.1.1)
User Datagram Protocol, Src Port: radius (1812), Dst Port: nfs (2049)
    Source port: radius (1812)
    Destination port: nfs (2049)
    Length: 107
    Checksum: 0xc062 (correct)
Radius Protocol
    Code: Access challenge (11)
    Packet identifier: 0x0 (0)
    Length: 99
    Authenticator: 0x782F035207756853938DA9B6E5F74B7D
    Attribute value pairs
```

```
        t:Reply Message(18) l:15, Value:"Hello, thomas"
        t:EAP Message(79) l:8
            Extensible Authentication Protocol
                Code: Request (1)
                Id: 1
                Length: 6
                Type: EAP-TLS [RFC2716] [Aboba] (13)
                Flags(0x20): Start
        t:Message Authenticator(80) l:18, Value:712A796A571F29FF4D50F0AB96AD6E17
        t:State(24) l:38, Value:78B0B7DEE98372ECFA4FA34436661EEB5555684081806C1F
                        DA012D4D40F81884A819EACA

0000  00 06 25 d7 b4 b3 00 e0 18 89 2d 16 08 00 45 00   ..%.......-...E.
0010  00 7f 00 18 40 00 40 11 b7 01 c0 a8 01 03 c0 a8   ....@.@.........
0020  01 01 07 14 08 01 00 6b c0 62 0b 00 00 63 78 2f   .......k.b...cx/
0030  03 52 07 75 68 53 93 8d a9 b6 e5 f7 4b 7d 12 0f   .R.uhS......K}..
0040  48 65 6c 6c 6f 2c 20 74 68 6f 6d 61 73 4f 08 01   Hello, thomasO..
0050  01 00 06 0d 20 50 12 71 2a 79 6a 57 1f 29 ff 4d   .... P.q*yjW.).M
0060  50 f0 ab 96 ad 6e 17 18 26 78 b0 b7 de e9 83 72   P....n..&x.....r
0070  ec fa 4f a3 44 36 66 1e eb 55 55 68 40 81 80 6c   ..O.D6f..UUh@..l
0080  1f da 01 2d 4d 40 f8 18 84 a8 19 ea ca            ...-M@.......
```

```
                                                                    # Frame 3
Frame 3 (304 bytes on wire, 304 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:53.817136000
    Time delta from previous packet: 0.075645000 seconds
    Time since reference or first frame: 0.076907000 seconds
    Frame Number: 3
    Packet Length: 304 bytes
    Capture Length: 304 bytes
Ethernet II, Src: 00:06:25:d7:b4:b3, Dst: 00:e0:18:89:2d:16
    Destination: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Source: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.1 (192.168.1.1),
                  Dst Addr: 192.168.1.3 (192.168.1.3)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 290
    Identification: 0xb4ba (46266)
    Flags: 0x04
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (0x11)
    Header checksum: 0x01bc (correct)
    Source: 192.168.1.1 (192.168.1.1)
    Destination: 192.168.1.3 (192.168.1.3)
User Datagram Protocol, Src Port: nfs (2049), Dst Port: radius (1812)
    Source port: nfs (2049)
    Destination port: radius (1812)
```

```
    Length: 270
    Checksum: 0x9bef (correct)
Radius Protocol
    Code: Access Request (1)
    Packet identifier: 0x0 (0)
    Length: 262
    Authenticator: 0x89D7D1B8000000000000000000000000
    Attribute value pairs
        t:User Name(1) l:8, Value:"thomas"
        t:NAS IP Address(4) l:6, Value:192.168.1.1
        t:Called Station Id(30) l:14, Value:"000625d7b4b5"
        t:Calling Station Id(31) l:14, Value:"00095b930108"
        t:NAS identifier(32) l:14, Value:"000625d7b4b5"
        t:NAS Port(5) l:6, Value:26
        t:Framed MTU(12) l:6, Value:1400
        t:State(24) l:38, Value:78B0B7DEE98372ECFA4FA34436661EEB5555684081
                        806C1FDA012D4D40F81884A819EACA
        t:NAS Port Type(61) l:6, Value:Wireless IEEE 802.11(19)
        t:EAP Message(79) l:112
            Extensible Authentication Protocol
              Code: Response (2)
              Id: 1
              Length: 110
              Type: EAP-TLS [RFC2716] [Aboba] (13)
              Flags(0x80): Length
              Length: 100
              Secure Socket Layer
                  TLS Record Layer: Client Hello
                      Content Type: Handshake (22)
                      Version: TLS 1.0 (0x0301)
                      Length: 95
                      Handshake Protocol: Client Hello
                          Handshake Type: Client Hello (1)
                          Length: 91
                          Version: TLS 1.0 (0x0301)
                          Random.gmt_unix_time: Mar 29, 2004 18:06:14.000000000
                          Random.bytes
                          Session ID Length: 0
                          Cipher Suites Length: 52
                          Cipher Suites (26 suites)
                            Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
                            Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
                            Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
                            Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
                            Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
                            Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
                            Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
                            Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
                            Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
                            Cipher Suite: TLS_DHE_DSS_WITH_RC4_128_SHA (0x0066)
                            Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
                            Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
                            Cipher Suite: TLS_DHE_DSS_EXPORT1024_WITH_DES_CBC_SHA (0x0063)
                            Cipher Suite: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x0062)
                            Cipher Suite: TLS_RSA_EXPORT1024_WITH_RC2_CBC_56_MD5 (0x0061)
                            Cipher Suite: TLS_DHE_RSA_WITH_DES_CBC_SHA (0x0015)
                            Cipher Suite: TLS_DHE_DSS_WITH_DES_CBC_SHA (0x0012)
                            Cipher Suite: TLS_RSA_WITH_DES_CBC_SHA (0x0009)
```

```
                                Cipher Suite: TLS_DHE_DSS_EXPORT1024_WITH_RC4_56_SHA (0x0065)
                                Cipher Suite: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x0064)
                                Cipher Suite: TLS_RSA_EXPORT1024_WITH_RC4_56_MD5 (0x0060)
                                Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA (0x0014)
                                Cipher Suite: TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA (0x0011)
                                Cipher Suite: TLS_RSA_EXPORT_WITH_DES40_CBC_SHA (0x0008)
                                Cipher Suite: TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (0x0006)
                                Cipher Suite: TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x0003)
                              Compression Methods Length: 1
                              Compression Methods (1 method)
                                  Compression Method: null (0)
      t:Message Authenticator(80) l:18, Value:43F6117D12786C5266D54828531ED957

0000  00 e0 18 89 2d 16 00 06 25 d7 b4 b3 08 00 45 00   ....-...%.....E.
0010  01 22 b4 ba 40 00 40 11 01 bc c0 a8 01 01 c0 a8   ."..@.@.........
0020  01 03 08 01 07 14 01 0e 9b ef 01 00 01 06 89 d7   ................
0030  d1 b8 00 00 00 00 00 00 00 00 00 00 00 00 01 08   ................
0040  74 68 6f 6d 61 73 04 06 c0 a8 01 01 1e 0e 30 30   thomas........00
0050  30 36 32 35 64 37 62 34 62 35 1f 0e 30 30 30 39   0625d7b4b5..0009
0060  35 62 39 33 30 31 30 38 20 0e 30 30 30 36 32 35   5b930108 .000625
0070  64 37 62 34 62 35 05 06 00 00 00 1a 0c 06 00 00   d7b4b5..........
0080  05 78 18 26 78 b0 b7 de e9 83 72 ec fa 4f a3 44   .x.&x.....r..O.D
0090  36 66 1e eb 55 55 68 40 81 80 6c 1f da 01 2d 4d   6f..UUh@..l...-M
00a0  40 f8 18 84 a8 19 ea ca 3d 06 00 00 00 13 4f 70   @.......=.....Op
00b0  02 01 00 6e 0d 80 00 00 00 64 16 03 01 00 5f 01   ...n.....d...._.
00c0  00 00 5b 03 01 40 68 49 76 0a ee df 95 4f 10 1e   ..[..@hIv....O..
00d0  b9 12 58 cb e6 03 a3 97 e1 02 b9 d3 e5 ac fe 41   ..X............A
00e0  3d fe 0c b9 c9 00 00 34 00 39 00 38 00 35 00 16   =......4.9.8.5..
00f0  00 13 00 0a 00 33 00 32 00 2f 00 66 00 05 00 04   .....3.2./.f....
0100  00 63 00 62 00 61 00 15 00 12 00 09 00 65 00 64   .c.b.a.......e.d
0110  00 60 00 14 00 11 00 08 00 06 00 03 01 00 50 12   .`............P.
0120  43 f6 11 7d 12 78 6c 52 66 d5 48 28 53 1e d9 57   C..}.xlRf.H(S..W

                                                                        # Frame 4-12

Frame 4 (1177 bytes on wire, 1177 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:53.823386000

Frame 5 (200 bytes on wire, 200 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.081679000

Frame 6 (1169 bytes on wire, 1169 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.107631000

Frame 7 (1514 bytes on wire, 1514 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.260084000

Frame 8 (132 bytes on wire, 132 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.260101000

Frame 10 (950 bytes on wire, 950 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.521336000

Frame 11 (204 bytes on wire, 204 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.538882000

Frame 12 (200 bytes on wire, 200 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.794271000
```

```
                                                                    # Frame 13
Frame 13 (217 bytes on wire, 217 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:54.795575000
    Time delta from previous packet: 0.001304000 seconds
    Time since reference or first frame: 1.055346000 seconds
    Frame Number: 13
    Packet Length: 217 bytes
    Capture Length: 217 bytes
Ethernet II, Src: 00:e0:18:89:2d:16, Dst: 00:06:25:d7:b4:b3
    Destination: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Source: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.1.3 (192.168.1.3),
                  Dst Addr: 192.168.1.1 (192.168.1.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 203
    Identification: 0x001d (29)
    Flags: 0x04
        0... = Reserved bit: Not set
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (0x11)
    Header checksum: 0xb6b0 (correct)
    Source: 192.168.1.3 (192.168.1.3)
    Destination: 192.168.1.1 (192.168.1.1)
User Datagram Protocol, Src Port: radius (1812), Dst Port: nfs (2049)
    Source port: radius (1812)
    Destination port: nfs (2049)
    Length: 183
    Checksum: 0x9f6a (correct)
Radius Protocol
    Code: Access Accept (2)
    Packet identifier: 0x0 (0)
    Length: 175
    Authenticator: 0x462A565C73193281A1692B77302C1B95
    Attribute value pairs
        t:Reply Message(18) l:15, Value:"Hello, thomas"
        t:Vendor Specific(26) l:58, Vendor:Microsoft(311)
            t:MS MPPE Recv Key(17) l:52,
    Value:C08262B3B28CCAD83AEE20F90518192917F64260AD44E503
          B0EA87D4E27BCCD134DA9098D621772AB5D83E20DA73F4D4DF16
        t:Vendor Specific(26) l:58, Vendor:Microsoft(311)
            t:MS MPPE Send Key(16) l:52,
    Value:C815D0B1EC6FA0C1A4AE263D65E9B0B02720654073DE9D51
          45264638BAC1A40006A6985B5EDCBFD0224EA02D6912FFCB9E8C
        t:EAP Message(79) l:6
            Extensible Authentication Protocol
                Code: Success (3)
                Id: 5
                Length: 4
```

```
        t:Message Authenticator(80) l:18, Value:DF4E7F7DC6108C881AD4A289F3F1CE60

0000  00 06 25 d7 b4 b3 00 e0 18 89 2d 16 08 00 45 00   ..%.......-...E.
0010  00 cb 00 1d 40 00 40 11 b6 b0 c0 a8 01 03 c0 a8   ....@.@.........
0020  01 01 07 14 08 01 00 b7 9f 6a 02 00 00 af 46 2a   .........j....F*
0030  56 5c 73 19 32 81 a1 69 2b 77 30 2c 1b 95 12 0f   V\s.2..i+w0,....
0040  48 65 6c 6c 6f 2c 20 74 68 6f 6d 61 73 1a 3a 00   Hello, thomas.:.
0050  00 01 37 11 34 c0 82 62 b3 b2 8c ca d8 3a ee 20   ..7.4..b.....:.
0060  f9 05 18 19 29 17 f6 42 60 ad 44 e5 03 b0 ea 87   ....)..B`.D.....
0070  d4 e2 7b cc d1 34 da 90 98 d6 21 77 2a b5 d8 3e   ..{..4....!w*..>
0080  20 da 73 f4 d4 df 16 1a 3a 00 00 01 37 10 34 c8    .s.....:...7.4.
0090  15 d0 b1 ec 6f a0 c1 a4 ae 26 3d 65 e9 b0 b0 27   ....o....&=e...'
00a0  20 65 40 73 de 9d 51 45 26 46 38 ba c1 a4 00 06    e@s..QE&F8.....
00b0  a6 98 5b 5e dc bf d0 22 4e a0 2d 69 12 ff cb 9e   ..[^..."N.-i....
00c0  8c 4f 06 03 05 00 04 50 12 df 4e 7f 7d c6 10 8c   .O.....P..N.}...
00d0  88 1a d4 a2 89 f3 f1 ce 60                        ........`

                                                                        # Frame 14
Frame 14 (42 bytes on wire, 42 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:58.740289000
    Time delta from previous packet: 3.944714000 seconds
    Time since reference or first frame: 5.000060000 seconds
    Frame Number: 14
    Packet Length: 42 bytes
    Capture Length: 42 bytes
Ethernet II, Src: 00:e0:18:89:2d:16, Dst: 00:06:25:d7:b4:b3
    Destination: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Source: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Type: ARP (0x0806)
Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    Sender MAC address: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Sender IP address: 192.168.1.3 (192.168.1.3)
    Target MAC address: 00:00:00:00:00:00 (00:00:00_00:00:00)
    Target IP address: 192.168.1.1 (192.168.1.1)

0000  00 06 25 d7 b4 b3 00 e0 18 89 2d 16 08 06 00 01
0010  08 00 06 04 00 01 00 e0 18 89 2d 16 c0 a8 01 03
0020  00 00 00 00 00 00 c0 a8 01 01

                                                                        # Frame 15
Frame 15 (60 bytes on wire, 60 bytes captured)
    Arrival Time: Mar 29, 2004 18:56:58.740641000
    Time delta from previous packet: 0.000352000 seconds
    Time since reference or first frame: 5.000412000 seconds
    Frame Number: 15
    Packet Length: 60 bytes
    Capture Length: 60 bytes
Ethernet II, Src: 00:06:25:d7:b4:b3, Dst: 00:e0:18:89:2d:16
    Destination: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Source: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Type: ARP (0x0806)
    Trailer: 000000000000000000000000000D24B...
Address Resolution Protocol (reply)
```

```
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (0x0002)
    Sender MAC address: 00:06:25:d7:b4:b3 (LinksysG_d7:b4:b3)
    Sender IP address: 192.168.1.1 (192.168.1.1)
    Target MAC address: 00:e0:18:89:2d:16 (AsustekC_89:2d:16)
    Target IP address: 192.168.1.3 (192.168.1.3)

0000  00 e0 18 89 2d 16 00 06 25 d7 b4 b3 08 06 00 01
0010  08 00 06 04 00 02 00 06 25 d7 b4 b3 c0 a8 01 01
0020  00 e0 18 89 2d 16 c0 a8 01 03 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00 d2 4b ad e8
```

Anhang B - Cisco Aironet Paketdump

Der folgende Paketdump wurde mir freundlicherweise zur Verfügung gestellt und zeigt eine erfolgreich verlaufene Authentifizierung mit LEAP.

		Source	Destination		
1	0.000000	AironetW_59:d9:a4	AironetW_5b:37:af	EAPOL	Start
2	0.001945	AironetW_5b:37:af	AironetW_59:d9:a4	EAP	Request, Identity [RFC2284]
3	0.010643	AironetW_59:d9:a4	AironetW_5b:37:af	EAP	Response, Identity [RFC2284]
4	0.016225	AironetW_5b:37:af	AironetW_59:d9:a4	EAP	Request, EAP-Cisco Wireless (LEAP) [Norman]
5	0.022071	AironetW_59:d9:a4	AironetW_5b:37:af	EAP	Response, EAP-Cisco Wireless (LEAP) [Norman]
6	0.027397	AironetW_5b:37:af	AironetW_59:d9:a4	EAP	Success
7	0.028270	AironetW_59:d9:a4	AironetW_5b:37:af	EAP	Request, EAP-Cisco Wireless (LEAP) [Norman]
8	0.036652	AironetW_5b:37:af	AironetW_59:d9:a4	EAP	Response, EAP-Cisco Wireless (LEAP) [Norman]
9	0.037001	AironetW_5b:37:af	AironetW_59:d9:a4	EAPOL	Key
10	0.037315	AironetW_5b:37:af	AironetW_59:d9:a4	EAPOL	Key
11	0.037680	AironetW_5b:37:af	AironetW_59:d9:a4	IEEE 802	Data

Challenge vom Server	af e8 11 f2 ae 94 8b db
Challenge Response vom Client	5b 79 da b8 bf 72 ed 43 4e bca8 a7 84 46 6b ff b2 8f 6e 94 28 0c 91 8d
Challenge vom Client	b1 b6 27 55 b1 26 52 55
Challenge Response vom Server	77 33 71 35 87 3f 73 c6 c6 6e f1 73 86 65 c1 1f e1 fa b4 90 98 7a 23 90

```
Frame 1 (36 bytes on wire, 36 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.856343000
    Time delta from previous packet: 0.000000000 seconds
    Time since reference or first frame: 0.000000000 seconds
    Frame Number: 1
    Packet Length: 36 bytes
    Capture Length: 36 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0108 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x1
            DS status: Frame is entering DS (To DS: 1  From DS: 0) (0x01)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 314
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    Destination address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1292
Logical-Link Control
    DSAP: SNAP (0xaa)
```

```
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Cisco IOS 9.0 Compatible (0x0000f8)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Start (1)
    Length: 0

0000  08 01 3a 01 00 40 96 5b 37 af 00 40 96 59 d9 a4   ..:..@.[7..@.Y..
0010  00 40 96 5b 37 af c0 50 aa aa 03 00 00 f8 88 8e   .@.[7..P........
0020  01 01 00 00                                       ....
```

Frame 2

```
Frame 2 (86 bytes on wire, 86 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.858288000
    Time delta from previous packet: 0.001945000 seconds
    Time since reference or first frame: 0.001945000 seconds
    Frame Number: 2
    Packet Length: 86 bytes
    Capture Length: 86 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x2
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1753
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
```

```
    Length: 50
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 1
        Length: 50
        Type: Identity [RFC2284] (1)
        Identity (45 bytes): \000networkid=RADIUS,nasid=AP350-5b37af,portid=0

0000  08 02 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   ..u..@.Y...@.[7.
0010  00 40 96 5b 37 af 90 6d aa aa 03 00 00 00 88 8e   .@.[7..m........
0020  01 00 00 32 01 01 00 32 01 00 6e 65 74 77 6f 72   ...2...2..networ
0030  6b 69 64 3d 52 41 44 49 55 53 2c 6e 61 73 69 64   kid=RADIUS,nasid
0040  3d 41 50 33 35 30 2d 35 62 33 37 61 66 2c 70 6f   =AP350-5b37af,po
0050  72 74 69 64 3d 30                                 rtid=0

                                                                        # Frame 3
Frame 3 (47 bytes on wire, 47 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.866986000
    Time delta from previous packet: 0.008698000 seconds
    Time since reference or first frame: 0.010643000 seconds
    Frame Number: 3
    Packet Length: 47 bytes
    Capture Length: 47 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0108 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x1
            DS status: Frame is entering DS (To DS: 1  From DS: 0) (0x01)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 314
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    Destination address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1294
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Cisco IOS 9.0 Compatible (0x0000f8)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 11
    Extensible Authentication Protocol
```

```
        Code: Response (2)
        Id: 1
        Length: 11
        Type: Identity [RFC2284] (1)
        Identity (6 bytes): RSAINI

0000  08 01 3a 01 00 40 96 5b 37 af 00 40 96 59 d9 a4   ..:..@.[7..@.Y..
0010  00 40 96 5b 37 af e0 50 aa aa 03 00 00 f8 88 8e   .@.[7..P........
0020  01 00 00 0b 02 01 00 0b 01 52 53 41 49 4e 49      .........RSAINI
```

Frame 4

```
Frame 4 (58 bytes on wire, 58 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.872568000
    Time delta from previous packet: 0.005582000 seconds
    Time since reference or first frame: 0.016225000 seconds
    Frame Number: 4
    Packet Length: 58 bytes
    Capture Length: 58 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x2
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1754
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 22
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 2
        Length: 22
        Type: EAP-Cisco Wireless (LEAP) [Norman] (17)
        Version: 1
```

```
        Reserved: 0
        Count: 8
        Peer Challenge [8] Random Value:"AFE811F2AE948BDB"
        Name (6 bytes): RSAINI

0000  08 02 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   ..u..@.Y...@.[7.
0010  00 40 96 5b 37 af a0 6d aa aa 03 00 00 00 88 8e   .@.[7..m........
0020  01 00 00 16 01 02 00 16 11 01 00 08 af e8 11 f2   ................
0030  ae 94 8b db 52 53 41 49 4e 49                     ....RSAINI
```

Frame 5

```
Frame 5 (74 bytes on wire, 74 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.878414000
    Time delta from previous packet: 0.005846000 seconds
    Time since reference or first frame: 0.022071000 seconds
    Frame Number: 5
    Packet Length: 74 bytes
    Capture Length: 74 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0108 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x1
            DS status: Frame is entering DS (To DS: 1  From DS: 0) (0x01)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 314
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    Destination address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1295
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Cisco IOS 9.0 Compatible (0x0000f8)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 38
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 2
        Length: 38
        Type: EAP-Cisco Wireless (LEAP) [Norman] (17)
        Version: 1
```

```
        Reserved: 0
        Count: 24
        Peer Challenge [8] Random Value:"5B79DAB8BF72ED434EBCA8A784466BFF..."
        Name (6 bytes): RSAINI

0000  08 01 3a 01 00 40 96 5b 37 af 00 40 96 59 d9 a4   ..:..@.[7..@.Y..
0010  00 40 96 5b 37 af f0 50 aa aa 03 00 00 f8 88 8e   .@.[7..P........
0020  01 00 00 26 02 02 00 26 11 01 00 18 5b 79 da b8   ...&...&....[y..
0030  bf 72 ed 43 4e bc a8 a7 84 46 6b ff b2 8f 6e 94   .r.CN....Fk...n.
0040  28 0c 91 8d 52 53 41 49 4e 49                     (...RSAINI

                                                                        # Frame 6
Frame 6 (40 bytes on wire, 40 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.883740000
    Time delta from previous packet: 0.005326000 seconds
    Time since reference or first frame: 0.027397000 seconds
    Frame Number: 6
    Packet Length: 40 bytes
    Capture Length: 40 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x2
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1755
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 4
    Extensible Authentication Protocol
        Code: Success (3)
        Id: 3
        Length: 4
```

```
0000  08 02 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   ..u..@.Y...@.[7.
0010  00 40 96 5b 37 af b0 6d aa aa 03 00 00 00 88 8e   .@.[7..m........
0020  01 00 00 04 03 03 00 04                           ........
```

Frame 7

```
Frame 7 (58 bytes on wire, 58 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.884613000
    Time delta from previous packet: 0.000873000 seconds
    Time since reference or first frame: 0.028270000 seconds
    Frame Number: 7
    Packet Length: 58 bytes
    Capture Length: 58 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0108 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x1
            DS status: Frame is entering DS (To DS: 1  From DS: 0) (0x01)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 314
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    Destination address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1296
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Cisco IOS 9.0 Compatible (0x0000f8)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 22
    Extensible Authentication Protocol
        Code: Request (1)
        Id: 3
        Length: 22
        Type: EAP-Cisco Wireless (LEAP) [Norman] (17)
        Version: 1
        Reserved: 0
        Count: 8
        Peer Response [24] NtChallengeResponse(B1B62755B1265255)
        Name (6 bytes): RSAINI

0000  08 01 3a 01 00 40 96 5b 37 af 00 40 96 59 d9 a4   ..:..@.[7..@.Y..
```

```
0010  00 40 96 5b 37 af 00 51 aa aa 03 00 00 f8 88 8e   .@.[7..Q........
0020  01 00 00 16 01 03 00 16 11 01 00 08 b1 b6 27 55   ..............'U
0030  b1 26 52 55 52 53 41 49 4e 49                     .&RURSAINI
```

Frame 8

```
Frame 8 (74 bytes on wire, 74 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.892995000
    Time delta from previous packet: 0.008382000 seconds
    Time since reference or first frame: 0.036652000 seconds
    Frame Number: 8
    Packet Length: 74 bytes
    Capture Length: 74 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x2
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1756
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: EAP Packet (0)
    Length: 38
    Extensible Authentication Protocol
        Code: Response (2)
        Id: 3
        Length: 38
        Type: EAP-Cisco Wireless (LEAP) [Norman] (17)
        Version: 1
        Reserved: 0
        Count: 24
        Peer Response [24] NtChallengeResponse(77337135873F73C6C66EF1738665C11F...)
        Name (6 bytes): RSAINI

0000  08 02 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   ..u..@.Y...@.[7.
```

```
0010  00 40 96 5b 37 af c0 6d aa aa 03 00 00 00 88 8e   .@.[7..m........
0020  01 00 00 26 02 03 00 26 11 01 00 18 77 33 71 35   ...&...&....w3q5
0030  87 3f 73 c6 c6 6e f1 73 86 65 c1 1f e1 fa b4 90   .?s..n.s.e......
0040  98 7a 23 90 52 53 41 49 4e 49                     .z#.RSAINI
```

Frame 9

```
Frame 9 (93 bytes on wire, 93 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.893344000
    Time delta from previous packet: 0.000349000 seconds
    Time since reference or first frame: 0.037001000 seconds
    Frame Number: 9
    Packet Length: 93 bytes
    Capture Length: 93 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x2
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1757
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Key (3)
    Length: 57
    Descriptor Type: RC4 Descriptor (1)
    Key Length: 13
    Replay Counter: 5348024631492621
    Key IV: 66332BF770940EAE23B076CB4BF7189C
    Key Index: broadcast, index 0
    0... .... = Key Type: Broadcast
    .000 0000 = Index Number: 0
    Key Signature: 727856C67CF6CDFD8213A09F201DB764
    Key: 6688C6CCA9138AE8752B361119

0000  08 02 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   ..u..@.Y...@.[7.
```

```
0010  00 40 96 5b 37 af d0 6d aa aa 03 00 00 00 88 8e   .@.[7..m........
0020  01 03 00 39 01 00 0d 00 13 00 00 04 69 00 0d 66   ...9........i..f
0030  33 2b f7 70 94 0e ae 23 b0 76 cb 4b f7 18 9c 00   3+.p...#.v.K....
0040  72 78 56 c6 7c f6 cd fd 82 13 a0 9f 20 1d b7 64   rxV.|....... ..d
0050  66 88 c6 cc a9 13 8a e8 75 2b 36 11 19            f.......u+6..
```

Frame 10

```
Frame 10 (80 bytes on wire, 80 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.893658000
    Time delta from previous packet: 0.000314000 seconds
    Time since reference or first frame: 0.037315000 seconds
    Frame Number: 10
    Packet Length: 80 bytes
    Capture Length: 80 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x0208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x2
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = WEP flag: WEP is disabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1758
Logical-Link Control
    DSAP: SNAP (0xaa)
    IG Bit: Individual
    SSAP: SNAP (0xaa)
    CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: 802.1X Authentication (0x888e)
802.1x Authentication
    Version: 1
    Type: Key (3)
    Length: 44
    Descriptor Type: RC4 Descriptor (1)
    Key Length: 13
    Replay Counter: 5348024631492622
    Key IV: 6FA377E7119C5D5C7BC36184353871D4
    Key Index: unicast, index 3
    1... .... = Key Type: Unicast
    .000 0011 = Index Number: 3
    Key Signature: EFD1F8AC4E1339C9F1698583C4B788C0
[Malformed Packet: EAPOL]
```

```
0000  08 02 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   ..u..@.Y...@.[7.
0010  00 40 96 5b 37 af e0 6d aa aa 03 00 00 00 88 8e   .@.[7..m........
0020  01 03 00 2c 01 00 0d 00 13 00 00 04 69 00 0e 6f   ...,........i..o
0030  a3 77 e7 11 9c 5d 5c 7b c3 61 84 35 38 71 d4 83   .w...]\{.a.58q..
0040  ef d1 f8 ac 4e 13 39 c9 f1 69 85 83 c4 b7 88 c0   ....N.9..i......
```

Frame 11

```
Frame 11 (90 bytes on wire, 90 bytes captured)
    Arrival Time: Nov 12, 2002 15:27:00.894023000
    Time delta from previous packet: 0.000365000 seconds
    Time since reference or first frame: 0.037680000 seconds
    Frame Number: 11
    Packet Length: 90 bytes
    Capture Length: 90 bytes
IEEE 802.11
    Type/Subtype: Data (32)
    Frame Control: 0x4208 (Normal)
        Version: 0
        Type: Data frame (2)
        Subtype: 0
        Flags: 0x42
            DS status: Frame is exiting DS (To DS: 0  From DS: 1) (0x02)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .1.. .... = WEP flag: WEP is enabled
            0... .... = Order flag: Not strictly ordered
    Duration: 117
    Destination address: 00:40:96:59:d9:a4 (AironetW_59:d9:a4)
    BSS Id: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Source address: 00:40:96:5b:37:af (AironetW_5b:37:af)
    Fragment number: 0
    Sequence number: 1759
    WEP parameters
        Initialization Vector: 0x640009
        Key: 3
        WEP ICV: 0x2f9d6f12 (not verified)
Data (58 bytes)

0000  08 42 75 00 00 40 96 59 d9 a4 00 40 96 5b 37 af   .Bu..@.Y...@.[7.
0010  00 40 96 5b 37 af f0 6d 09 00 64 c0 77 a6 0e c7   .@.[7..m..d.w...
0020  6d b2 1d 92 dd 0f e9 9d 1a c8 49 25 16 ce 7f f0   m.........I%....
0030  45 55 3c 44 9f b2 85 78 57 3b df 32 30 4a 11 35   EU<D...xW;.20J.5
0040  b6 09 07 1c de eb 38 55 31 59 77 0e 78 88 ff 5c   ......8U1Yw.x..\
0050  24 41 cd 4a 52 1f 2f 9d 6f 12                     $A.JR./.o.
```

Anhang C - Association Phase in WLAN 802.11i

In 802.11i WLAN müssen, bevor eine gewöhnliche Kommunikation stattfinden kann, mehrere einleitende Verhandlungen stattfinden. Am Anfang steht die "Discovery und Association"-Phase (Abb. 5.4). Client und Access Point verständigen sich über die von ihnen unterstützten Sicherheitsfunktionen (security capabilities).

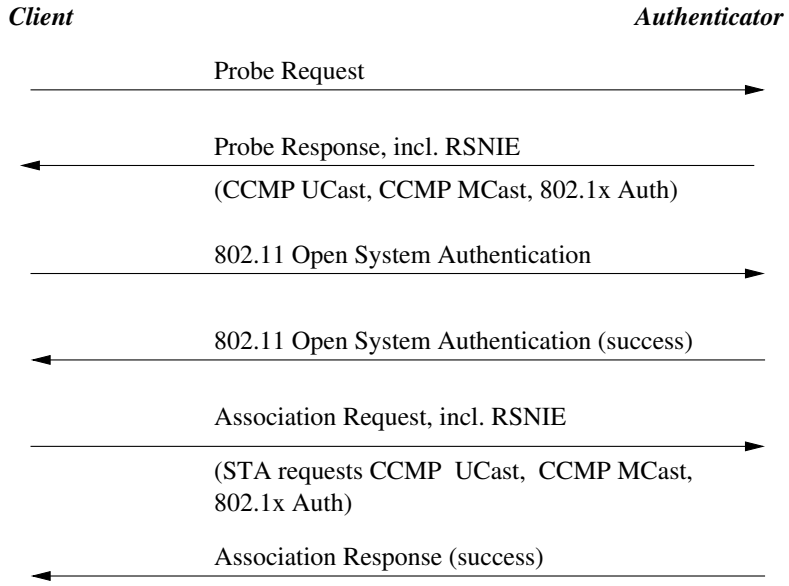


Abbildung 5.4: 802.11i Association Phase

Diese in 802.11i neu hinzugekommenen Informationen werden in Management Frames als sogenanntes RSNIE (RSN Information Element) mitgeführt, sind also in (vgl. [802.11i], 7.3.2.25) Beacon Frames, (Re-)Association Requests sowie Probe Response Frames zu finden. Der Standard verweist darauf, dass diese Informationen nicht als verbindlich zu interpretieren sind, sondern eher zur Steigerung der Performance verwandt werden sollten. Indem der Client das RSNIE ausliest, kann er die Forderungen vom Access Point schon im Vorhinein auf Übereinstimmung mit seiner Security Policy prüfen. Die von ihm zurückgeschickte Auswahl in Form eines RSNIE zieht der Access Point heran, um zu entscheiden, ob eine Assoziation mit dem Client aufgebaut oder diese abgelehnt werden soll.

Das RSN Information Element beinhaltet die zur Verfügung stehenden Authentication Suites, die verfügbaren Unicast Ciphersuites und Multicast Ciphersuites. Im Falle von alternativen Suites kann der Client eine Auswahl treffen. Eine Alternative bedeutet zum Beispiel, wenn der Access Point TKIP, CCMP und WEP-104 für Verschlüsselung anbietet.

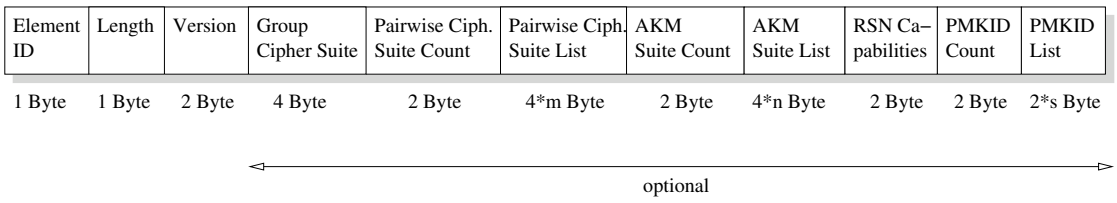


Abbildung 5.5: RSN Information Element

Einige Hinweise zu den Feldern: Die Element ID soll 30 Hex, also 48 sein. Das Länge Feld beinhaltet die Anzahl Bytes vom Version Feld an gerechnet. Die momentane RSN Version ist mit "1" vorgegeben. AKM steht für Authentication and Key Management. Ein Suite Selector hat das Format OUI (3 Byte) | Suite Type (1 Byte).

OUI	Suite Type	Bedeutung
00 0F AC	0	Use Group Cipher Suite
00 0F AC	1	WEP-40
00 0F AC	2	TKIP
00 0F AC	3	reserviert
00 0F AC	4	CCMP (Vorgabe in einer RSNA)
00 0F AC	5	WEP-104
00 0F AC	6-255	reserviert

Tabelle 5.3: Cipher Suite Selectors

Anstatt die weiteren Felder zu beschreiben, soll ein repräsentatives Beispiel der Illustration dienen: Der Access Point bietet 802.1X Authentifizierung, CCMP Pairwise and Group Cipher Suites an. Dahingegen sind WEP-40, WEP-104 und TKIP nicht erlaubt. Das RSNIE hat unter diesen Bedingungen den folgenden Aufbau:

30	information element id, 48 expressed as Hex value (Vorgabe)
14	length in octets, 20 expressed as Hex value
01 00	Version 1
00 0F AC 04	CCMP as group cipher suite
01 00	pairwise cipher suite count
00 0F AC 04	CCMP as pairwise cipher suite
01 00	authentication count
00 0F AC 01	**802.1X authentication**
00 00	No capabilities

Tabelle 5.4: RSNIE Beispiel

Die Authentifizierung, wie sie in dieser Arbeit behandelt wird, setzt die in in Tabelle 5.4 hervorgehobenen 4 Byte für 802.1x Authentifizierung voraus.

Anhang D - PPP EAP Typen (IANA)

Type	Beschreibung	Quelle
1	Identity	[RFC2284]
2	Notification	[RFC2284]
3	Nak (Response only)	[RFC2284]
4	MD5-Challenge	[RFC2284]
5	One Time Password (OTP)	[RFC2289]
6	Generic Token Card	[RFC2284]
7		
8		
9	RSA Public Key Authentication	[Whelan]
10	DSS Unilateral	[Nace]
11	KEA	[Nace]
12	KEA-VALIDATE	[Nace]
13	EAP-TLS	[Aboba]
14	Defender Token (AXENT)	[Rosselli]
15	RSA Security SecurID EAP	[Asnes, Liberman]
16	Arcot Systems EAP	[Jerdonek]
17	EAP-Cisco Wireless	[Norman]
18	Nokia IP smart card authentication	[Haverinen]
19	SRP-SHA1 Part 1	[Carlson]
20	SRP-SHA1 Part 2	[Carlson]
21	EAP-TTLS	[Funk]
22	Remote Access Service	[Fields]
23	UMTS Authentication and Key Argreement	[Haverinen]
24	EAP-3Com Wireless	[Young]
25	PEAP	[Palekar]
26	MS-EAP-Authentication	[Palekar]
27	Mutual Authentication w/Key Exchange (MAKE)	[Berrendonner]
28	CRYPTOCard	[Webb]
29	EAP-MSCHAP-V2	[Potter]
30	DynamID	[Merlin]
31	Rob EAP	[Ullah]
32	SecurID EAP	[Josefsson, Liberman]
33	MS-Authentication-TLV	[Palekar]
34	SentriNET	[Kelleher]
35	EAP-Actiontec Wireless	[Chang]
36	Cogent Systems Biometrics Authentication EAP	[Xiong]
37	AirFortress EAP	[Hibbard]
38	EAP-HTTP Digest	[Tavakoli]
39	SecureSuite EAP	[Clements]
40	DeviceConnect EAP	[Pitard]
41	EAP-SPEKE	[Zick]
42	EAP-MOBAC	[Rixom]
43	EAP-FAST	[Cam-Winget]
44	ZoneLabs EAP (ZLXEAP)	[Bogue]