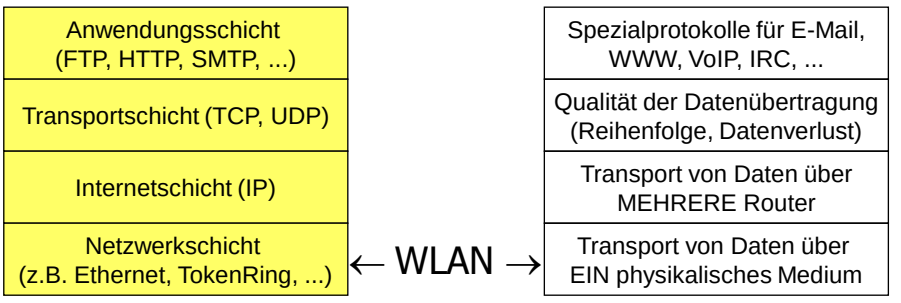
Netzsicherheit

Teil 3: Wireless LAN

Prof. Dr. Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

Das TCP/IP-Schichtenmodell

- Datentransport wird durch eine Kombination von Protokollen ermöglicht, die sich jeweils um Teilaufgaben kümmern.
- Das TCP/IP-Schichtenmodell beschreibt diese Aufgabenteilung für das Internet.



Gliederung

1. Einführung WLAN-Technologie
2. WEP: Wired Equivalent Privacy
3. What's Wrong With WEP? (Borisov, Goldberg, Wagner)
4. Die Stromchiffre RC4
5. Key Scheduling Weaknesses in RC4 (Fluhrer, Mantin, Shamir)

Netwerkschicht: Ethernet, WLAN & Co

- Übertragungsprotokolle für Teilnetze mit gleicher Technologie
  - Ethernet: ursprünglich Broadcast-Netz
  - PPP: Punkt-zu-Punkt-Verbindung
  - WLAN: Broadcast-Netz
  - DVB: unidirektionales Broadcast-Netz mit Fehlerkorrektur
- Analogien Ethernet – WLAN
  - klassisches Ethernet (mit Hubs, ohne Switches): drahtgebundener Broadcast
  - WLAN: funkbasierter Broadcast
  - Datenverkehr kann in beiden Fällen mitgelesen werden (Sniffing, Wardriving)

WLAN: IEEE 802.11

- Erster Standard 1997: IEEE 802.11
  - 2,4 GHz Industry, Scientific an Medical (ISM) Band
  - 1 oder 2 Mbps
  - Frequency Hopping Spread Spectrum oder Direct Sequence Spread Spectrum
- Mittlerweile gibt es IEEE 802.11 a,b,c,d,e,f,g,h,i,j,k,l,m,n
- 2 Modi:
  - Infrastructure: Alle Hosts sind über Access Point verbunden
  - Ad-Hoc: Je zwei Hosts können direkt kommunizieren
- Standard beinhaltet „Wired Equivalent Privacy“ (WEP), vollständig gebrochen:
  - Walker (Oct 2000),
  - Borisov, Goldberg, Wagner(Jan 2001),
  - Fluhrer, Mantin, Shamir (Aug 2001)

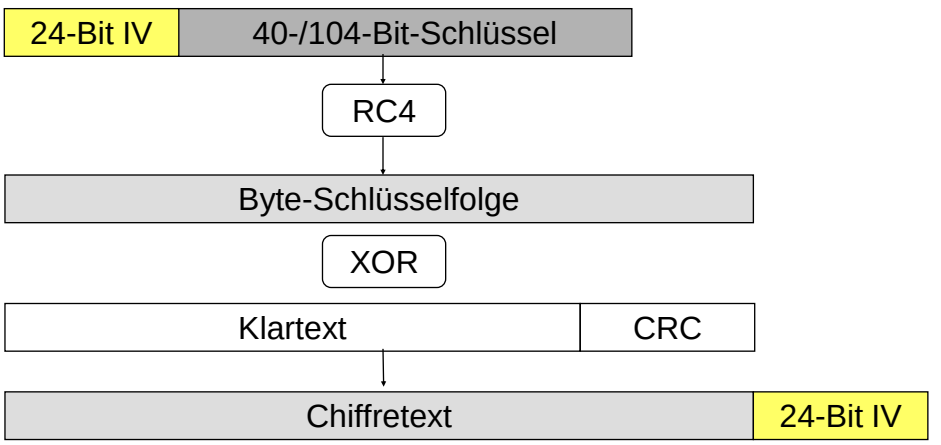
WLAN: Sicherheitsmaßnahmen ohne WEP

- Service Set IDentification (SSID)
  - SSIDs sollen ein Funknetz in logische Einheiten unterteilen
  - Ein Access Point akzeptiert alle Hosts mit vorgegebener SSID
  - SSIDs werden im Klartext übertragen
  - Fazit: Wird als Sicherheitsmaßnahme eingesetzt (Open Systems Authentication), ist aber keine.
- MAC Adress Filtering
  - Für jeden Access Point wird eine Liste mit zugelassenen MAC-Adressen erstellt
  - Schwer zu administrieren
  - MAC Spoofing ist möglich

Gliederung

1. Einführung WLAN-Technologie
2. WEP: Wired Equivalent Privacy
3. What's Wrong With WEP? (Borisov, Goldberg, Wagner)
4. Die Stromchiffre RC4
5. Key Scheduling Weaknesses in RC4 (Fluhrer, Mantin, Shamir)

WEP



WEP (2)

- CRC-Prüfsumme c(M): Einfacher, fehlererkennender Code, kein MAC!
- Die Daten M bilden zusammen mit der CRC-Prüfsumme c(M) den Klartext $P = M \| c(M)$
- Der geheime symmetrische Schlüssel k bildet zusammen mit dem Initialisierungsvektor IV die Eingabe für den RC4-Algorithmus
- RC4 ist eine Stromchiffre, bei der der Schlüsselstrom in 8-Bit-Blöcken erzeugt wird.
  - Der Schlüsselstrom wird mit dem Klartext P bitweise XOR-verknüpft. Das Ergebnis ist der Chiffretext.
- Der Chiffretext wird zusammen mit IV auf dem Funkkanal übertragen.
- Es gibt kein Schlüsselmanagement im WEP-Standard!
  - In der Regel wurden alle Hosts eines WLAN mit demselben RC4-Schlüssel konfiguriert!

Gliederung

1. Einführung WLAN-Technologie
2. WEP: Wired Equivalent Privacy
3. What's Wrong With WEP? (Borisov, Goldberg, Wagner)
4. Die Stromchiffre RC4
5. Key Scheduling Weaknesses in RC4 (Fluhrer, Mantin, Shamir)

WEP: Keystream Reuse

- Der Schlüsselstrom k_{IV} zur Verschlüsselung von WEP-Paketen hängt nur von dem Schlüssel k und dem IV ab:
  $k_{IV} = RC4(k,IV)$
- IV wird im Klartext übertragen, k ist fest
- Folgerung 1 (gleicher IV):
  - Falls $C1 = P1 \oplus k_{IV}$
    und $C2 = P2 \oplus k_{IV}$
    dann
- Folgerung 2:

WEP: Keystream Reuse

- Der Schlüsselstrom k_{IV} zur Verschlüsselung von WEP-Paketen hängt nur von dem Schlüssel k und dem IV ab:
  $k_{IV} = RC4(k,IV)$
- IV wird im Klartext übertragen, k ist fest
- Folgerung 1 (gleicher IV):
  - Falls $C1 = P1 \oplus k_{IV}$
    und $C2 = P2 \oplus k_{IV}$
    dann $C1 \oplus C2 = P1 \oplus k_{IV} \oplus P2 \oplus k_{IV} = P1 \oplus P2$
    - Es gibt Methoden, um P1 und P2 aus $P1 \oplus P2$ zu berechnen.
- Folgerung 2:

WEP: Keystream Reuse

- Der Schlüsselstrom k_{IV} zur Verschlüsselung von WEP-Paketen hängt nur von dem Schlüssel k und dem IV ab:
  $k_{IV} = RC4(k,IV)$
- IV wird im Klartext übertragen, k ist fest
- Folgerung 1 (gleicher IV):
  - Falls $C1 = P1 \oplus k_{IV}$
    und $C2 = P2 \oplus k_{IV}$
    dann $C1 \oplus C2 = P1 \oplus k_{IV} \oplus P2 \oplus k_{IV} = P1 \oplus P2$
  - Es gibt Methoden, um P1 und P2 aus $P1 \oplus P2$ zu berechnen.
- Folgerung 2:
  - Decryption Dictionaries mit 2^{24} Einträgen für festen Schlüssel k (egal welcher Länge) möglich!

WEP: Keystream Reuse (2)

Wann ist Keystream Reuse möglich?

- Annahmen:
  - A: Für jeden Hosts ein eigener Schlüssel
  - B: Nur ein Schlüssel k für das gesamte Netzwerk
- Spätestens nach 2^{24} Nachrichtenpaketen...
  - A: ... eines Hosts!
  - B: ... des Netzwerks!! Bei einem AP mit 1500 Byte-Paketen und 11 Mbps ist das nach 5 Stunden der Fall. Dann müsste eigentlich der Schlüssel (manuell!) gewechselt werden.
- Mit Wahrscheinlichkeit 1/2 schon nach ca. 5000 Paketen (Geburtstagsparadoxon!)
- Mit noch größerer Wahrscheinlichkeit, wenn ein Gerät nach Reset den IV immer von 0 hochzählt (mehrfach in der Praxis aufgetreten).

WEP: Modifikation von Nachrichten

- Die CRC-Prüfsumme ist linear, d.h. es gilt

$$c(X \oplus Y) = c(X) \oplus c(Y)$$

- Angriff: Ändere die verschlüsselte Nachricht M zu M‘.
  - Berechne $D = M' \ominus M$ ($= M' \oplus M$)
  - Berechne $D \| c(D)$
  - Berechne
  - C‘ wird vom Empfänger zu M‘ entschlüsselt

WEP: Modifikation von Nachrichten

- Die CRC-Prüfsumme ist linear, d.h. es gilt

$$c(X \oplus Y) = c(X) \oplus c(Y)$$

- Angriff: Ändere die verschlüsselte Nachricht M zu M‘.
  - Berechne $D = M' \ominus M$ ($= M' \oplus M$)
  - Berechne $D \| c(D)$
  - Berechne $C' = C \oplus D \| c(D)$
    $= k_{IV} \oplus M \| c(M) \oplus D \| c(D)$
    $= k_{IV} \oplus (M \oplus D) \| (c(M) \oplus c(D))$
    $= k_{IV} \oplus M' \| c(M')$
  - C‘ wird vom Empfänger zu M‘ entschlüsselt

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

WEP: Modifikation von Nachrichten (2)

Anwendungen:

- Verschlüsselung beliebiger Nachrichten an Empfänger:
  - Sende bekannten Plaintext P=M||c(M) an den Empfänger (z.B. eine SPAM-Mail)
  - Zeichne das verschlüsselte Paket, insbesondere den IV, auf
  - Modifiziere P zu beliebigem P' und sende das Paket
  - Funktioniert, weil alte IVs weiter verwendet werden dürfen ohne Alarm auszulösen
- Entschlüsselung durch IP Redirection
  - Modifiziere die IP-Adresse $IP_{original}$ in M (teilweise oder ganz bekannt) in $IP_{Angreifer}$. AP entschlüsselt, Firewalls lassen IP von innen nach außen durch.
  - Problem: Prüfsummen in IP- und TCP-Header

WEP: Modifikation von Nachrichten (2)

Anwendungen:

- Entschlüsselung durch IP Redirection
  - Problem: Prüfsummen in IP- und TCP-Header
    - Prüfsumme des Originalpakets ist bekannt: ✓
    - Differenz der Prüfsummen kann mit großer W. geraten werden
    - Ändere anderes Headerfeld so ab, dass die Prüfsumme gleich bleibt
- TCP-Ack-Attacke
  - Ändere ein paar Bits in M ab
  - Wenn TCP-Prüfsumme weiterhin korrekt, wird ein kurzes ACK gesendet (sonst: "silently discard packet")
  - Dies liefert Information über den Klartext

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

Gliederung

1. Einführung WLAN-Technologie
2. WEP: Wired Equivalent Privacy
3. What's Wrong With WEP? (Borisov, Goldberg, Wagner)
4. Die Stromchiffre RC4
5. Key Scheduling Weaknesses in RC4 (Fluhrer, Mantin, Shamir)

RC4

- Ron Rivest 1987
- geheim bis 1994
- Besteht aus zwei Phasen
  - In der Key Setup-Phase (KSA) wird mit Hilfe des Schlüssels ein interner Startzustand aus den $2^8! \approx 2^{1700}$ möglichen Zuständen ausgewählt
  - In der Ausgabephase (PRGA) werden aus dem Startzustand Nachfolgezustände generiert und dabei jeweils ein Byte ausgegeben.

RC4

KSA(K)
Initialization ($N=2^8$):
    For i = 0 ... N-1
        S[i] = i
    j = 0

Scrambling
    For i = 0 ... N-1
        j = j + S[i] + K[i mod L]
        Swap(S[i] , S[j])

PRGA(K)
Initialization:
    i = 0
    j = 0

Generation Loop
    i = i+1
    j = j + S[i]
    Swap(S[i] , S[j])
    Output Z = S[S[i] + S[j]]

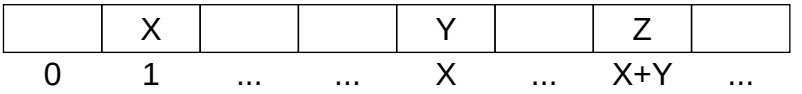
RC4 mit bekanntem IV

- Idee: Beobachte nur das 1. Byte der Ausgabe von PRGA
- Dieses Byte wird aus dem Startzustand wie folgt erzeugt:
  - i = 0+1 = 1
  - j = 0 + S[1] =: X
  - Swap(S[1] , S[X])
  - Output Z = S[S[1] + S[X]]
- Fazit: Wenn wir den Wert der Speicherstellen 1 (=X) und X (=Y) kennen,

	X			Y		Z	
0	1	...	...	X	...	X+Y	...

RC4 mit bekanntem IV

- Idee: Beobachte nur das 1. Byte der Ausgabe von PRGA
- Dieses Byte wird aus dem Startzustand wie folgt erzeugt:
  - i = 0+1 = 1
  - j = 0 + S[1] =: X
  - Swap(S[1] , S[X])
  - Output Z = S[S[1] + S[X]]
- Fazit: Wenn wir den Wert der Speicherstellen 1 (=X) und X (=Y) kennen, dann ist das erste ausgegebene Byte des Schlüsselstroms (=Z) der Wert an der Speicherstelle X+Y.
- Ziel: Wert an Speicherstelle X+Y liefert uns Informationen zum nächsten Schlüsselbyte



Gliederung

1. Einführung WLAN-Technologie
2. WEP: Wired Equivalent Privacy
3. What's Wrong With WEP? (Borisov, Goldberg, Wagner)
4. Die Stromchiffre RC4
5. Key Scheduling Weaknesses in RC4 (Fluhrer, Mantin, Shamir)

RC4 mit IV vor dem geheimen Schlüssel

Ziel: Berechne den geheimen Schlüssel Byte für Byte

- Annahme:
  - Die ersten Bytes K[3], ..., K[A+2] des geheimen Schlüssels sind bereits berechnet,
  - gesucht ist K[A+3]
- Suche WEP-Pakete mit IVs der Form

  (A+3, N-1, X)

  für ca. 60 verschiedene Werte von X.

RC4 mit IV vor dem geheimen Schlüssel

Ziel: Berechne den geheimen Schlüssel Byte für Byte

- Annahme:
  - Die ersten Bytes K[3], ..., K[A+2] des geheimen Schlüssels sind bereits berechnet,
  - gesucht ist K[A+3]
- Suche WEP-Pakete mit IVs der Form

  Länge des Arrays

  (A+3, N-1, X)

  für ca. 60 verschiedene Werte von X.
- Da wir jetzt die ersten A+3 Schlüsselbytes

  A+3, N-1, X, K[3], ..., K[A+2]

  kennen, können wir KSA von RC4 bis zu einem gewissen Punkt nachvollziehen.

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt 0: i = 0, j = 0+S[0]+K[0] = A+3

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
0	1	2	3	...	...	A+2	A+3	...

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt 0: i = 0, j = 0+S[0]+K[0] = A+3

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	1	2	3	...	...	A+2	0	...

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt 0: i = 0, j = 0+S[0]+K[0] = A+3

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	1	2	3	...	...	A+2	0	...

- KSA Schritt 1: i = 1, j = (A+3)+S[1]+K[1] = A+3+1+N-1 = A+3

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	1	2	3	...	...	A+2	0	...

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt 0: i = 0, j = 0+S[0]+K[0] = A+3

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	1	2	3	...	...	A+2	0	...

- KSA Schritt 1: i = 1, j = (A+3)+S[1]+K[1] = A+3+1+N-1 = A+3

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	2	3	...	...	A+2	1	...

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt 2: i=2, j=(A+3)+S[2]+K[2]=A+3+2+X

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	2	3	...	...	A+2	1	...

- KSA Schritt 3 bis A+2:
  - J enthält jetzt den (zufälligen, aber bekannten) IV-Wert X, daher können alle weiteren Swap-Operationen als zufällig angesehen werden.
  - Nach Schritt A+2 kennt der Angreifer den Wert j_{A+2} und die genauen Werte der Permutation $S_{A+2}[0]$, ..., $S_{A+2}[N-1]$
  - Wenn $S_{A+2}[0] \neq A+3$ und $S_{A+2}[1] \neq 0$, also nicht mehr mit dem Bild oben übereinstimmen, wird die Berechnung abgebrochen und der IV verworfen.

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt 2: i=2, j=(A+3)+S[2]+K[2]=A+3+2+X

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	A+5+X	3	...	...	A+2	1	...

- KSA Schritt 3 bis A+2:
  - J enthält jetzt den (zufälligen, aber bekannten) IV-Wert X, daher können alle weiteren Swap-Operationen als zufällig angesehen werden.
  - Nach Schritt A+2 kennt der Angreifer den Wert j_{A+2} und die genauen Werte der Permutation $S_{A+2}[0]$, ..., $S_{A+2}[N-1]$
  - Wenn $S_{A+2}[0] \neq A+3$ und $S_{A+2}[1] \neq 0$, also nicht mehr mit dem Bild oben übereinstimmen, wird die Berechnung abgebrochen und der IV verworfen.

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt A+3: i = A+3, $j_{A+3} = j_{A+2}+S_{A+2}[A+3]+K[A+3]$

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	$S_{A+2}[2]$	$S_{A+2}[3]$	...	...	...	$S_{A+2}[j_{A+3}]$	...

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt A+3: i = A+3, $j_{A+3} = j_{A+2}+S_{A+2}[A+3]+K[A+3]$
  - Angreifer kennt j_{A+2}, $S_{A+2}[0]$, ..., $S_{A+2}[A+3]$, ..., $S_{A+2}[N-1]$

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	$S_{A+2}[2]$	$S_{A+2}[3]$	...	...	...	$S_{A+2}[j_{A+3}]$	...

RC4 mit IV vor dem geheimen Schlüssel

- KSA Schritt A+3: i = A+3, $j_{A+3} = j_{A+2}+S_{A+2}[A+3]+K[A+3]$
  - Angreifer kennt j_{A+2}, $S_{A+2}[0]$, ..., $S_{A+2}[A+3]$, ..., $S_{A+2}[N-1]$
  - Falls der Angreifer $S_{A+3}[A+3]=S_{A+2}[j_{A+3}]$ kennen würde:
    - Könnte er diesen Wert in $S_{A+2}[0]$, ..., $S_{A+2}[N-1]$ suchen und daraus j_{A+3} bestimmen.
    - Dann ist $K[A+3] = j_{A+3} - j_{A+2} - S_{A+2}[A+3]$
  - Frage: Wann ist das der Fall?

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	$S_{A+2}[2]$	$S_{A+2}[3]$	...	...	...	$S_{A+2}[j_{A+3}]$	...

RC4 mit IV vor dem geheimen Schlüssel

- Wann kennt ein Angreifer $S_{A+2}[j_{A+3}]$?
- Z.B. wenn die Elemente S[0]=A+3, S[1]=0 und S[A+3] von den nachfolgenden (N-1-A-3) Swap-Operationen nicht verändert werden.

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	$S_{A+2}[2]$	$S_{A+2}[3]$	...	...	...	$S_{A+2}[j_{A+3}]$	...

Jörg Schwenk
Lehrstuhl für Netz- und Datensicherheit

RC4 mit IV vor dem geheimen Schlüssel

- Wann kennt ein Angreifer $S_{A+2}[j_{A+3}]$?
- Z.B. wenn die Elemente S[0]=A+3, S[1]=0 und S[A+3] von den nachfolgenden (N-1-A-3) Swap-Operationen nicht verändert werden.
  - Denn dann gilt im 1. Schritt von PRGA(K):
    i = 0+1, j = 0+S[1]=0,
    z = S[S[1]+S[0]] = S[0 + A+3] = S[A+3] = $S_{A+2}[j_{A+3}]$
  - Dies ist für jeden passenden IV mit W. ≈ 0,05 der Fall, nach ca. 60 passenden IVs ist die W. > 0,5.

A+3	N-1	X	K[3]	...	...	K[A+2]	K[A+3]	...
0	1	2	...	...	...	A+2	A+3	...
A+3	0	$S_{A+2}[2]$	$S_{A+2}[3]$	...	...	...	$S_{A+2}[j_{A+3}]$	...

RC4 mit IV vor dem geheimen Schlüssel

Zusammenfassung des Angriffs:

- Bedingt durch die 802.11-Codierung ist das erste Byte des Klartextes eines WEP-Pakets immer bekannt
- Der Angriff von Fluhrer et. al. kann daher durchgeführt werden.
- Komplexität wächst nur linear mit der Schlüssellänge
- Implementiert in zahllosen Wardriving-Tools:
  http://www.wardrive.net/wardriving/tools/

Ausblick 1: Die Unsicherheit von WEP

Erste Gegenmaßnahme der Hersteller: Filtern der „weak IVs“, d.h. der IV-Werte, die für den oben beschriebenen Angriff verwendet werden.

- KoreK (2004):
  - Statistische Angriffe auf WEP, die keine „weak IVs“ mehr brauchen
  - Reduktion der Framezahl auf 500.000
- Klein (2007):
  - Statistischer Angriff auf WEP ohne „weak IVs“ anhand „ähnlicher“ Schlüssel
- Tews, Weinmann, Pyshkin (2007):
  - Variante des Angriffs von Klein, alle Schlüsselbytes werden parallel berechnet; aktiver Angriff, 85.000 Pakete

Ausblick 2: WPA 1/2

Temporal Key Integrity Protocol (TKIP):

- RC4 mit 48 Bit IV und insgesamt 128 Bit Schlüssellänge
- Ersatz der CRC-Prüfsumme durch MAC auf Basis der Chiffre Michael: Nur 20 Bit Sicherheit, heuristische Gegenmaßen erforderlich

Counter-Mode/CBC-MAC Protocol (CCMP): AES

- AES im Counter Mode
- 128-Bit-MAC auf Basis von AES-CBC, nur 64 Bit werden übertragen

Für beide Verfahren gibt es die beiden Key Agreement-Klassen:

- Pre-Shared Key (PSK)
- Extensible Authentication Protocol (EAP) / RADIUS: Intensive Standardisierung, auch unsichere Verfahren, aber auch SSL/TLS