

# Kapitel 12: Netzsicherheit - WLAN-Sicherheit (Schicht 2)



- WLAN: Eine kurze Einführung
- WLAN-Sicherheitsanforderungen und Mechanismen
- Wired Equivalent Privacy (WEP)
  - Authentisierung
  - Vertraulichkeit
  - Integrität
  - Autorisierung
  - Schwächen und Angriffe
- WiFi Protected Access (WPA)
  - Authentisierung mit 802.1X oder Preshared Keys (PSK)
  - Vertraulichkeit (TKIP)
  - TKIP-Schlüsselhierarchie
  - WPA- und TKIP-Sicherheit
- WPA 2



- WLAN standardisiert in IEEE 802.11x:

Standard	Frequenz [GHz]	maximaler Durchsatz [Mbit/s]
802.11	2,4	2
802.11a	5	54
802.11b	2,4	11
802.11g	2,4	54
802.11n	2,4 / 5	600
802.11ac (2014 verabschiedet)	5	1,69 Gbit/s (6,77 Gbit/s)

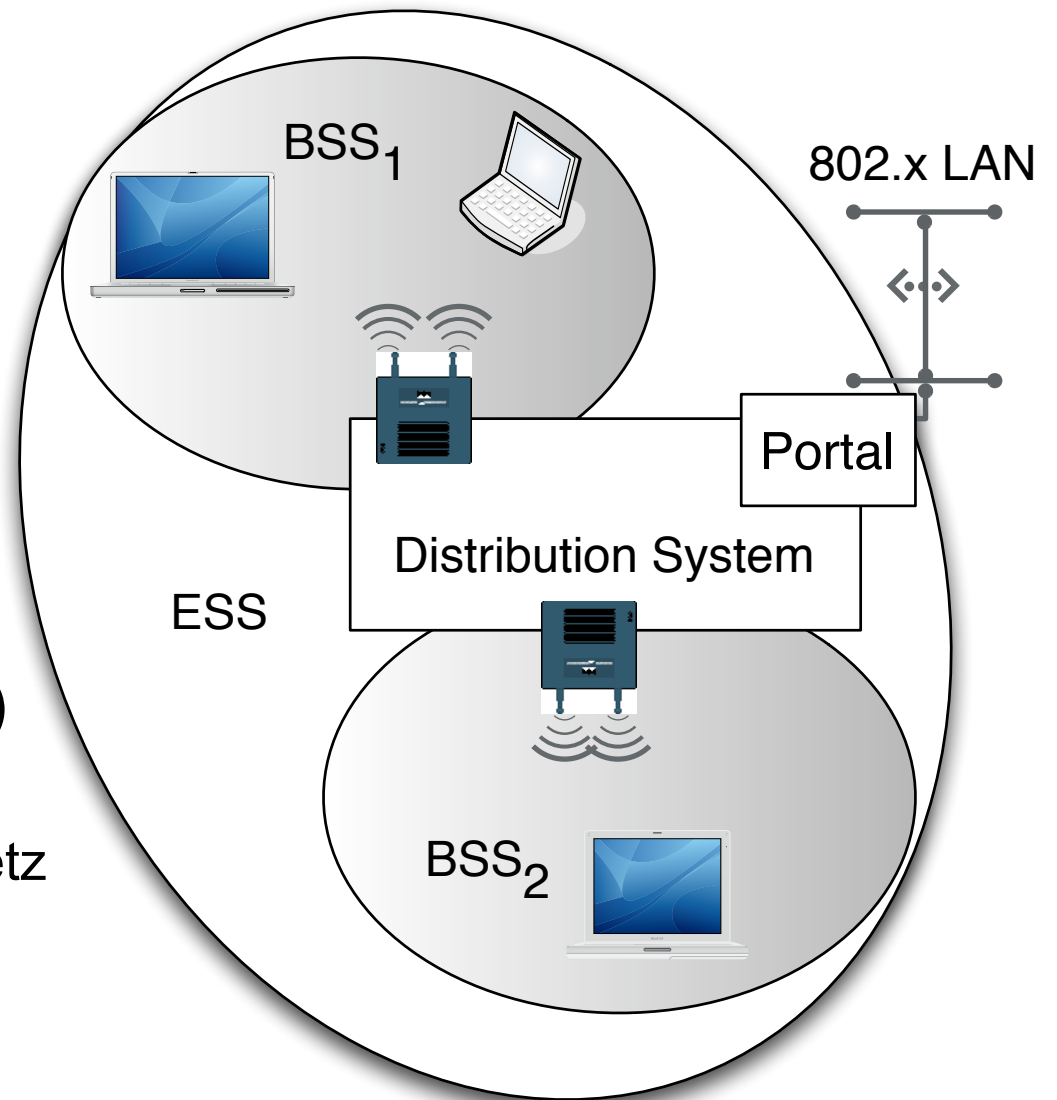
- Alle Geräte teilen sich die Bandbreite
- Maximaler Durchsatz praktisch nicht erreichbar (netto wird i.d.R. weniger als die Hälfte erreicht, z.B. 200-300 Mbit/s bei 802.11n)

- Derzeit leistungsfähigste Geräte im MWN:  
Alcatel-Lucent AP-325 und AP-275
- Dualband-AP, d.h. 2,4 GHz- und 5 GHz-Frequenzband
- Zwei Radios
- Durchsatz bei opt. Bedingungen 1.300 Mbit/s (AP-275 mit 802.11ac) bzw. 450 mbit/s brutto
- Controller basierte Lösung entwickelt von Aruba

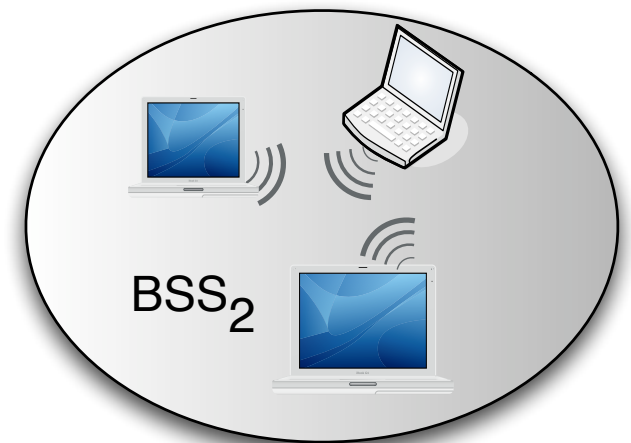


- Nutzungsstatistik installierter Access Points:  
<http://wlan.lrz.de/apstat/>

- Access Point (AP):  
Zugangsknoten zum WLAN
- Station (STA)
  - Gerät mit WLAN-Ausstattung
  - (Intelligenter) Client
- Basic Service Set (BSS)
  - Gruppe von STAs, die selbe Frequenz nutzen
- Extended Service Set (ESS)
  - logisches Netz aus mehreren BSS
  - wird gebildet durch Verbindungsnetz (Distribution System (DSS))
  - ESS wird durch SSID identifiziert
- Portal: Verbindung zu anderen Netzen



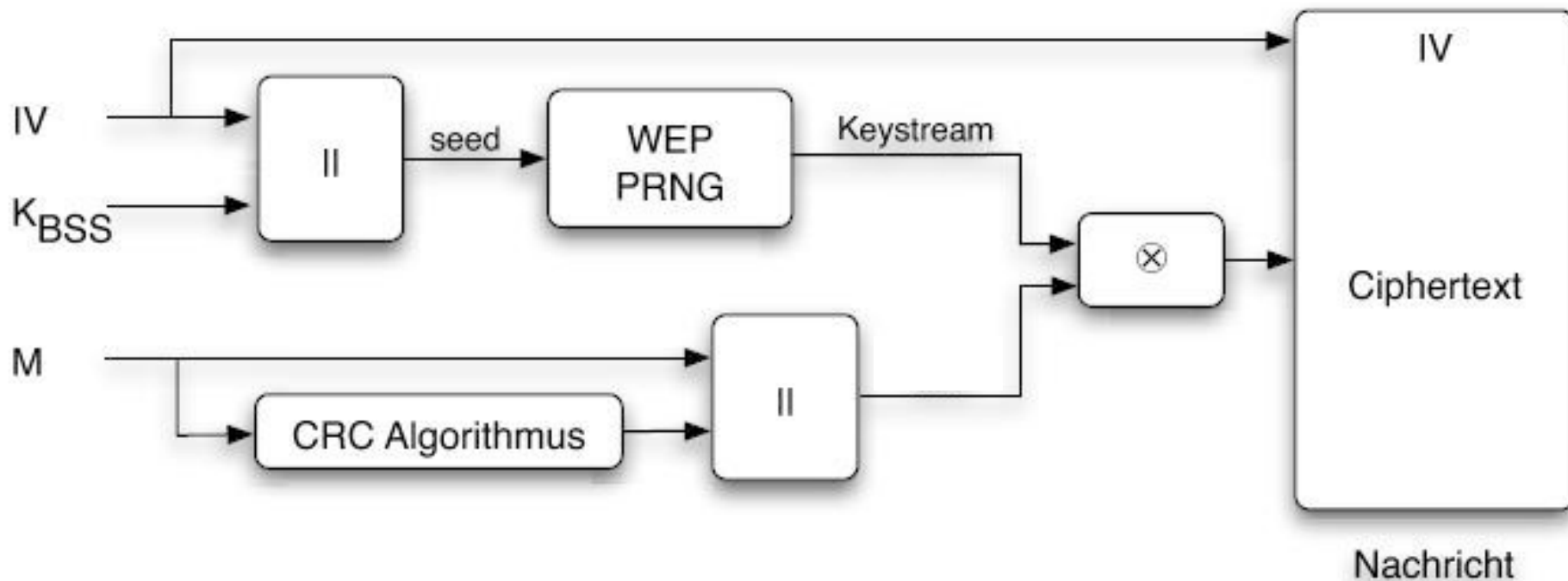
- Kein Access Point (AP) erforderlich
- Alle Stationen sind gleichberechtigt
- Basic Service Set (BSS)
  - Gruppe von STAs, die dieselbe Frequenz nutzen
  - Keine Kommunikation zwischen BSS möglich



- Mallet und Eve haben es im WLAN (wg. Funk) noch einfacher als in kabelgebundenen Netzen
- Sicherheitsanforderungen
  - Authentisierung der Teilnehmer
  - Zugangskontrolle zum Netz (Autorisierung)
  - Vertraulichkeit der Daten
  - Integrität der Daten
- Sicherheitsmechanismen
  - Wired Equivalent Privacy (WEP)
  - WiFi Protected Access (WPA)
  - WiFi Protected Access 2 (WPA2)
  - IEEE 802.11i (Standard, wegen Verspätung etablierte die Wi-Fi Alliance (Herstellerkonsortium) bereits WPA)
    - IEEE 802.11i D3.0 ist äquivalent zu WPA
    - IEEE 802.11i D9.0 ist äquivalent zu WPA2

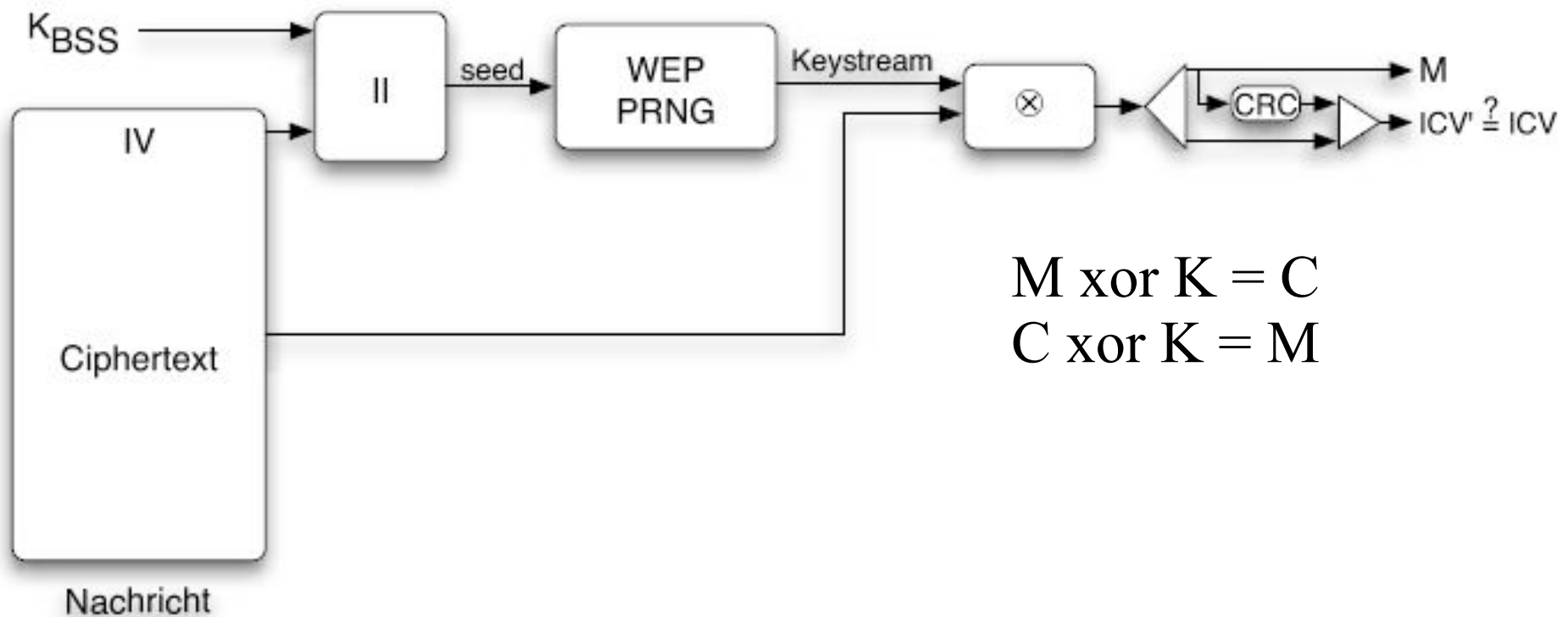
# Vertraulichkeit: Wired Equivalent Privacy (WEP)

- Klartext wird mit Bitstrom XOR-verknüpft
- Bitstrom wird mit RC4 als Pseudozufallszahlengenerator (WEP PRNG) erzeugt
  - Für jede Nachricht 24-bit Initialisierungsvektor (IV) konkateniert mit 40-bit WEP-Schlüssel als 64-bit Seed für PRNG
  - Nachricht konkateniert mit CRC wird mit dem Bitstrom XOR-verknüpft





- IV wird im Klartext mit jedem Chiffretext übertragen
  - Jeder, der  $K_{BSS}$  kennt, kann Keystream erzeugen und Nachricht entschlüsseln
  - Selbstsynchronisierung von WEP
- Entschlüsselung ist inverser Vorgang zur Verschlüsselung



- Cyclic Redundancy Check (CRC) ist ein Fehlererkennungscode
- Entwickelt, um Übertragungsfehler u.a. in Ethernet zu erkennen
- Mathematische Grundlagen:
  - Bit-String wird als Polynom mit Koeffizienten 0 und 1 aufgefasst
  - Nachricht  $M$  wird interpretiert als Polynom  $M(x)$
  - Berechnungen modulo 2; d.h. Addition und Subtraktion identisch mit XOR
- Berechnung des CRC-Werts von  $M(x)$  zur Integritätssicherung:
  - Einigung auf Generatorpolynom  $G(x)$  (i.d.R. standardisiert)
  - Sei  $n$  der Grad von  $G(x)$ , dann ist  $n+1$  die Länge des Bit-Strings von  $G(x)$
  - $M(x)$  wird durch  $G(x)$  geteilt
  - Teilungsrest  $M(x) \bmod G(x)$  ist CRC-Wert und wird an  $M$  angehängt
  - Empfänger berechnet: Gesamtnachricht  $(M(x) \mid \text{CRC}) \bmod G(x)$ 
    - = 0; Nachricht wurde bei der Übertragung nicht verändert  
(außer Änderung ist Vielfaches von  $G(x)$ )
    - $\neq 0$ ; Nachricht wurde verändert

- Einfach und billig in Hardware umzusetzen (32-bit Schieberegister)
- Gut geeignet für die Erkennung von „zufälligen“ Fehlern (z.B. bei Rauschen)
  - Ethernet
  - Festplatten-Datenübertragung
  - USB, Bluetooth, SD/MMC-Karten, ...
- Aber: CRC ist keine kryptographische Hashfunktion!
  - Andere (sinnvolle) Nachrichten mit selbem CRC-Wert können relativ einfach erzeugt werden
- Nur Fehlererkennung, keine Fehlerkorrektur möglich

## ■ Open System Authentication

- ❑ Entweder der AP verschlüsselt nicht: Dann keine Authentifizierung, jeder kann den AP nutzen
- ❑ Oder bei aktivierter WEP-Verschlüsselung: Wer den Schlüssel kennt, kann Daten übertragen

## ■ Shared Key Authentication

- ❑ 4-Way-Challenge-Response-Protokoll
- ❑ Basiert auf WEP-Verschlüsselung:
  1. STA sendet Authentication Request an AP
  2. AP sendet Challenge  $r$  im Klartext zurück
  3. STA verschlüsselt  $r$  und sendet  $WEP(r)$  zurück
  4. AP verifiziert

- Bei Open System Authentication ohne Verschlüsselung kann jeder senden
- Falls WEP aktiviert ist, kann nur senden, wer KBSS kennt
- Keine individuelle Benutzerauthentifizierung mittels WEP möglich
- Viele APs bieten zusätzlich MAC-adressbasierte Access Control Listen (ACLs)
  - Nur bekannte/freigeschaltete MAC Adressen dürfen senden, aber
    - MAC kann einfach mitgelesen werden
    - MAC kann einfach gefälscht werden

- WEP erfüllt KEINE der Sicherheitsanforderungen:
  
- Vertraulichkeit:
  - Schlüsselmanagement und Schlüssel sind ein Problem
  - WEP ist einfach zu brechen
  - Jeder der  $K_{BSS}$  kennt, kann alle damit verschlüsselten Nachrichten mitlesen
  
- Integrität
  - CRC ist kein geeignetes Verfahren zur Integritätssicherung bei absichtlicher Manipulation
  
- Authentisierung
  - basiert auf WEP
  
- Zugriffskontrolle
  - Keine individuelle Authentifizierung, somit generell nur rudimentäre Zugriffskontrolle möglich

- Standard legt kein Schlüsselmanagement fest
- „Out-of-Band“ Schlüsselverteilung erforderlich
  - Manuelles Schlüsselmanagement oft fehlerbehaftet
  - Schlüssel werden sehr selten gewechselt (in der Praxis eher nie)
  - Oft war per Default in Accesspoints die Open System Authentication ganz ohne Verschlüsselung aktiviert
  
- Schlüssellängen
  - WEP-40; 40 Bit Schlüssel (wegen Exportrestriktionen)
  - WEP-104; 104 Bit Schlüssel
    - Vom Benutzer z.B. in Form von 26 Hexziffern einzugeben
    - Somit mühsam/fehleranfällig und deshalb häufig sehr einfach gewählt
  - Aber selbst mit ausreichend langen Schlüsseln wäre WEP nicht sicher

- RC4 ist Stromchiffre, d.h. der selbe Seed sollte nicht wiederverwendet werden
  - IV soll dies verhindern
  - IV wird aber im Klartext mit übertragen
  - 24 Bit für den IV sind deutlich zu kurz
- Wiederverwendung des Keystream (bei gleichem IV)
  - Zwei Klartextnachrichten  $M_1$  und  $M_2$  mit Plaintext  $P_i = (M_i | CRC_i)$
  - Mit Ciphertext  $C_1 = P_1 \oplus RC4(IV_1, K_{BSS})$
  - und  $C_2 = P_2 \oplus RC4(IV_1, K_{BSS})$  gilt:
  - $C_1 \oplus C_2 = (P_1 \oplus RC4(IV_1, K_{BSS})) \oplus (P_2 \oplus RC4(IV_1, K_{BSS})) = P_1 \oplus P_2$
  - d.h. falls Angreifer  $M_1$  und  $C_1$  kennt, kann er  $P_2$  (somit  $M_2$ ) aus dem mitgehörten  $C_2$  berechnen, ohne  $K_{BSS}$  zu kennen (Known-Plaintext Angriff)
  - Known-Plaintext ist einfach zu erzeugen (Daten von außen schicken)



- Known-Plaintext Angriff: Mallet kennt M und C:  
 $C = RC4(IV, K_{BSS}) \oplus (M, CRC(M))$
- Damit kann Mallet den Key Stream berechnen:  
 $RC4(IV, K_{BSS}) = C \oplus (M, CRC(M))$
- Absichtliche Wiederverwendung alter IVs möglich:  
Mallet berechnet  
 $C' = RC4(IV, K_{BSS}) \oplus (M', CRC(M'))$   
und schickt  $(IV, C')$  an Bob
- Bob hält dies für ein gültiges Paket
  
- Wissen über verwendete höherliegende Protokolle erleichtert auch einen rein passiven Known-Plaintext Angriff:
  - Protokoll-Header, Adressen, Protokollprimitive sind Teile von M, meist an festen und bekannten Positionen

- CRC und RC4 sind linear
- Mallet fängt Nachricht von Alice an Bob ab:  $(IV, C)$  mit  $C = RC4(IV, K_{BSS}) \oplus (M, CRC(M))$
- Mallet verfälscht die Nachricht  $M$  zu Nachricht  $X$ :
  - Mallet wählt beliebige Nachricht  $M'$  mit derselben Länge
  - Mallet sendet Ciphertext  $C' = C \oplus (M', CRC(M')) =$   
 $RC4(IV, K_{BSS}) \oplus (M, CRC(M)) \oplus (M', CRC(M')) =$   
 $RC4(IV, K_{BSS}) \oplus (M \oplus M', CRC(M) \oplus CRC(M')) =$   
 $RC4(IV, K_{BSS}) \oplus (M \oplus M', CRC(M \oplus M')) =$   
 $RC4(IV, K_{BSS}) \oplus (X, CRC(X))$
- Mallet kennt Inhalt von  $X$  nicht, da er  $M$  nicht kennt
- Aber: Eine „1“ an Position  $n$  in  $M'$  führt zu gekipptem Bit an Position  $n$  in  $X$ ; Mallet kann kontrollierte Änderungen in  $M$  durchführen. Beispiel: Zieladresse von IP-Paketen ändern

- Papier von Fluhrer, Mantin und Shamir; 2001:
  - Grosse Zahl unsicherer Schlüssel wurden identifiziert, kleine Zahl von Bits reicht, um die meisten Output-Bits zu berechnen
  - Schwäche: IV wird mit KBSS konkateniert; IV im Klartext übertragen
  - $K_{BSS}$  bleibt relativ lange konstant, IV wechselt
  - Passive Ciphertext-Only Attack:
    - Eve muss 4 bis 6 Millionen Pakete mithören
    - Dies dauert nur wenige Minuten (ggf. Traffic stimulieren)
    - Abhängigkeit von der Schlüssellänge (40 oder 104 Bit) ist nur linear
  
- Klein zeigt 2005, dass es stärkere Korrelationen zwischen Keystream und Schlüssel gibt und verbessert diesen Angriff weiter

- Artikel von Tews, Weinmann, Pyshkin, TU Darmstadt, 2007
- Aktiver Angriff
- Nutzt ARP-Request- und ARP-Reply-Pakete
  - Feste Länge der Pakete
  - Über Länge der Frames sind die verschlüsselten ARP Pakete erkennbar
  - Die ersten 16 Byte des ARP Paketes sind vorhersagbar
    - 8 Byte LLC Header (AA AA 03 00 00 00 08 06) gefolgt von
    - 8 Byte ARP Header:
      - 00 01 08 00 06 04 00 01 für ARP Request
      - 00 01 08 00 06 04 00 02 für ARP Response
  - XOR Verknüpfung abgehörter Pakete mit dieser Bytefolge liefert die ersten 16 Byte des Keystream
  - Wiedereinspielen abgehörter ARP Requests beschleunigt den Angriff
  - Erfolgsrate bei nur 40.000 Frames schon > 50 %
  - Erfolgsrate bei 85.000 Frames rund 95 %

- WEP ist **NICHT** sicher
- WEP sollte **NICHT** verwendet werden
- Der Data Security Standard (DSS) der Payment Card Industry (PCI) verbietet die Nutzung von WEP im Rahmen jeglicher Kreditkarten-Datenverarbeitung seit Juli 2010

- WPA zur Verbesserung der Sicherheit eingeführt
- WEP-Hardware sollte weiter benutzbar bleiben
- Vertraulichkeit:
  - Temporal Key Integrity Protocol (TKIP)
  - Rekeying-Mechanismus zum automatischen Wechseln der Schlüssel
  - Hierarchie von Schlüsseln
- Integritätssicherung
  - TKIP Message Integrity Code - MIC (genannt „Michael“);  
zur Unterscheidung von MAC (Media Access Control)
  - Mit Schlüssel parametrisierte kryptographische Hash-Funktion
  - Verbessert ungeeigneten CRC-Mechanismus von WEP
- Authentisierung
  - Nach wie vor Möglichkeit für Pre-Shared Key (PSK)
  - Bietet aber auch 802.1X (insb. in großen IT-Infrastrukturen genutzt)

- TKIP verwendet Schlüsselhierarchie, um kurzlebige Schlüssel zu erzeugen
- Drei Hierarchiestufen (von unten nach oben):
  1. Temporäre Schlüssel (Temporal Key, TK)
    - In jede Richtung (AP zu STA, STA zu AP) eigene Schlüssel:
      - zur Verschlüsselung (128 Bit)
      - zur Integritätssicherung (64 Bit)
    - Erneuerung des Schlüsselmaterials durch `rekey key` Nachricht
    - `rekey key` Nachricht enthält Material, damit STA und AP neue Sitzungsschlüssel ableiten können; Nachricht verschlüsselt mit
  2. Pairwise Transient Key (PTK)
    - Sichern die Übertragung temporärer Schlüssel
    - 1 Schlüssel zur Sicherung des Schlüsselmaterials
    - 1 Schlüssel zur Sicherung der `rekey key` Nachricht

## 3. Pairwise Master Key (PMK)

- Höchster Schlüssel innerhalb der Hierarchie
- Erzeugt vom 802.1X Authentication Server und vom AP an STA weitergereicht
- Individuell pro Endgerät (AP)
- Falls 802.1X Setup „zu komplex“; Preshared Keys möglich (d.h. in der Praxis: Passwörter)
- Master Key wird zur Sicherung der key-encryption Keys genutzt
- Damit Aufbau einer Sitzungsstruktur möglich; von der Authentisierung über 802.1X bis
  - Widerruf des Schlüssels
  - Ablauf des Schlüssels
  - STA verliert Kontakt zum AP
  
- **Achtung: Kompromittierung des Master Key führt zur Kompromittierung der gesamten Hierarchie!**



- Aus IEEE 802.11i-2004 (geht über reines TKIP hinaus)
- hier Verwendung von 802.1X

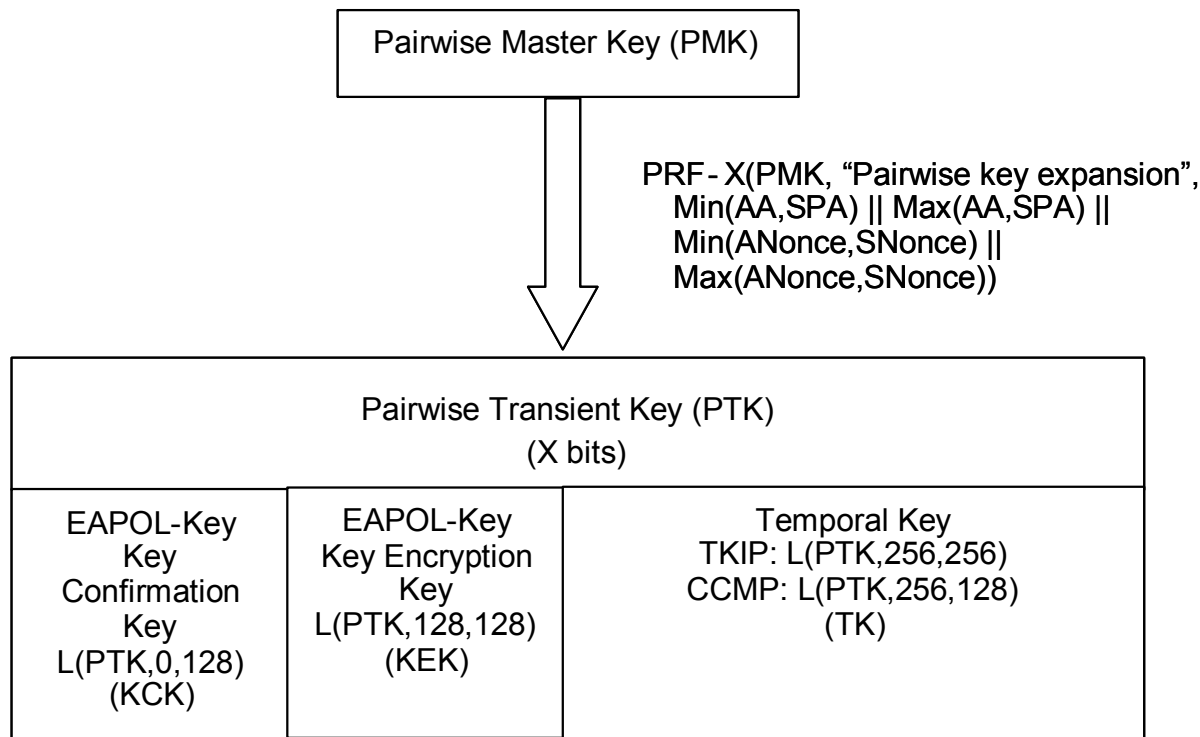
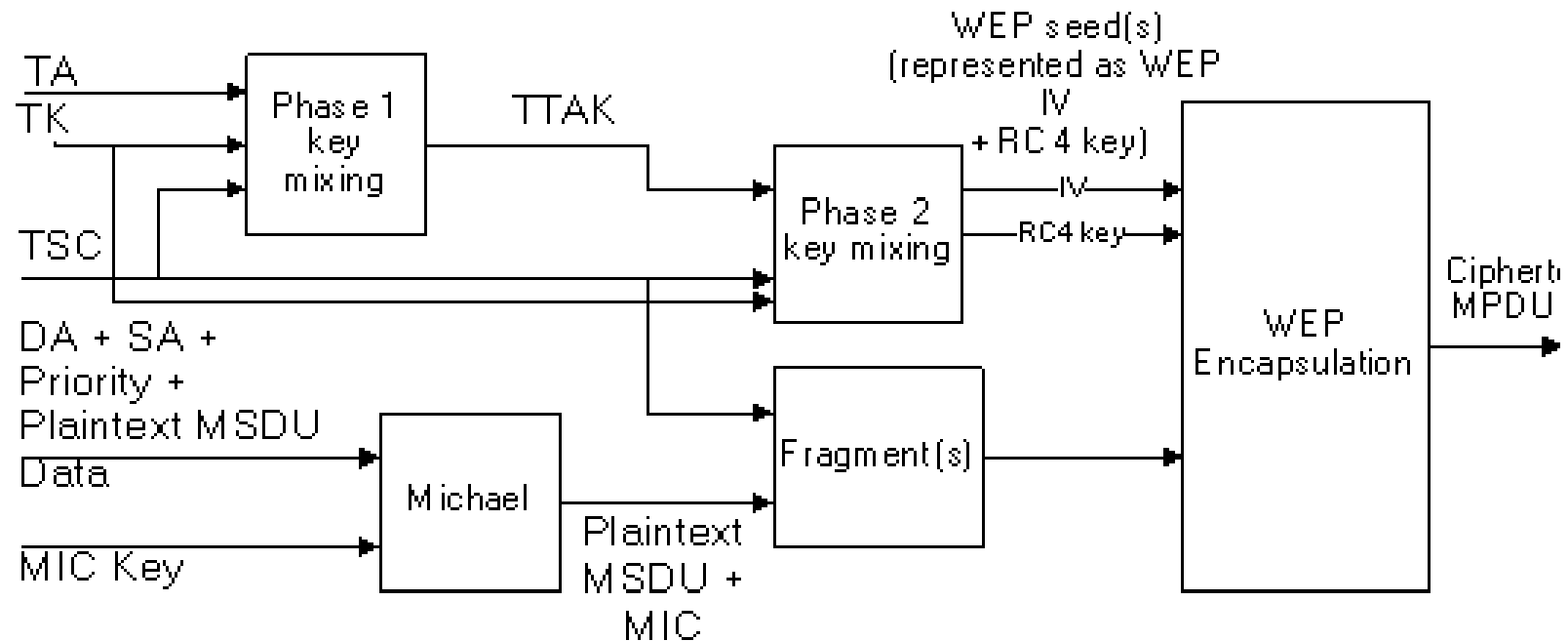


Figure 43s—Pairwise key hierarchy

- **PRF** Pseudo Random Function
- **AA** Authenticator Address
- **SPA** Supplicant Address
- **EAPOL** EAP over LAN
- **KCK** Key Confirmation Key (Integritätssicherung)
- **KEK** Key Encryption Key
- **L(x,0,128)** Teilstring ab Bit 0 mit Länge von 128
- **X(x) = L(x,0,512)** bei TKIP;  
**L(x,0,384)** bei CCMP

- CCMP ist Bestandteil von WPA2 (später, S. 39)
- PRF: Pseudo Random Function zur Schlüsselableitung (vgl. PKCS#5 oder RFC2898)

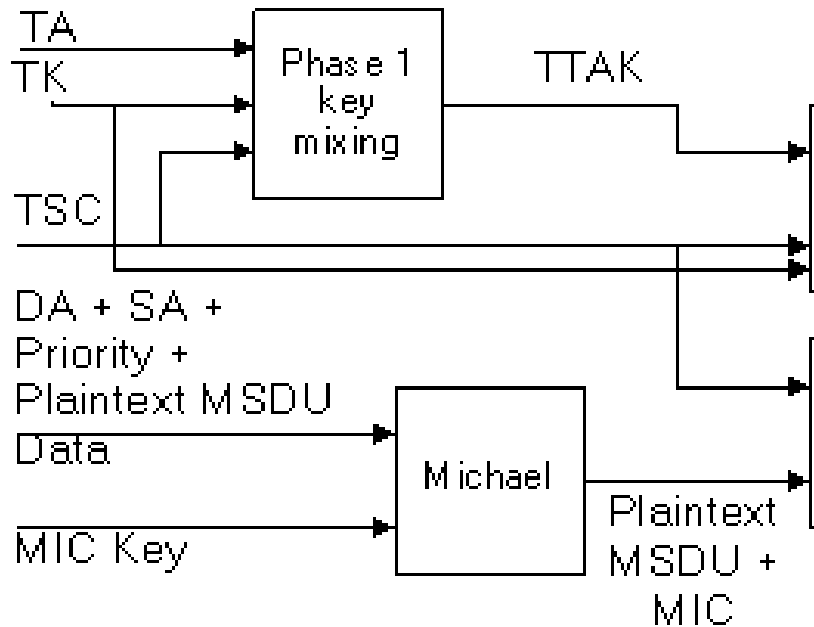
## ■ Aus IEEE 802.1i-2004



- TA Transmitter Address
- TK Temporal Key
- TSC TKIP Sequence Counter
- DA Destination Address
- SA Source Address

- MSDU MAC Service Data Unit
- MPDU Message Protocol Data Unit
- TTAk TKIP Mixed Address and Key
- MIC Message Integrity Code

## ■ Aus IEEE 802.11-2004



■ Kein wirklich neues Verfahren; soll nur Schwächen beseitigen

### ■ Phase 1 Key Mixing

■ TKIP Mixed Address and Key:  
TTAK = Phase1(TA, TK, TSC)

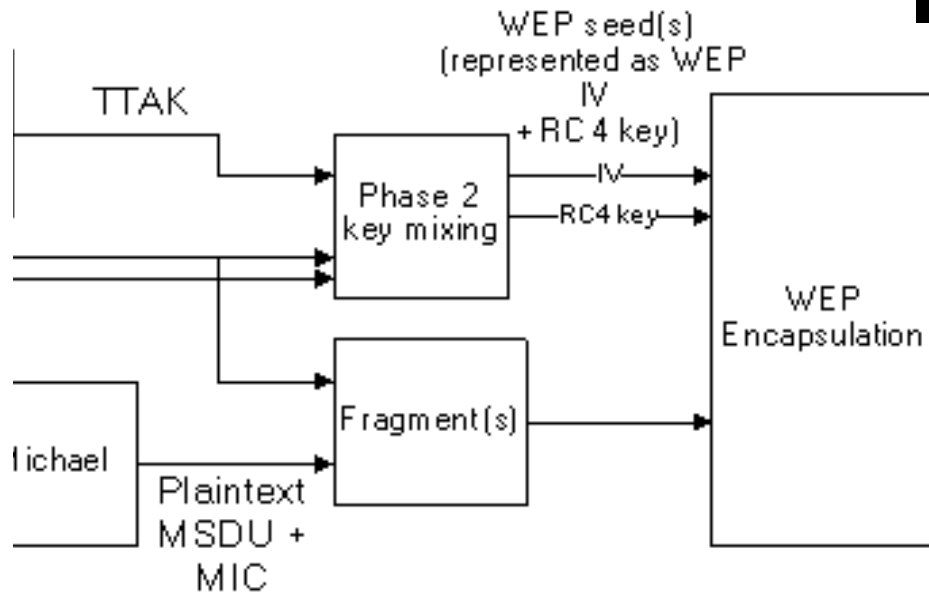
■ Phase1 ist nichtlineare Funktion mit XOR-Operationen, bitweiser UND-Operation sowie einer Verkürzungsfunktion

■ TA verhindert, dass zwei STAs denselben Schlüssel erhalten

■ TSC als Sequenznummer für Nachrichtenblöcke (MPDUs)

- TA Transmitter Address
- TK Temporal Key
- TSC TKIP Sequence Counter
- DA Destination Address
- SA Source Address

- Aus [IEEE 802.1i-2004]

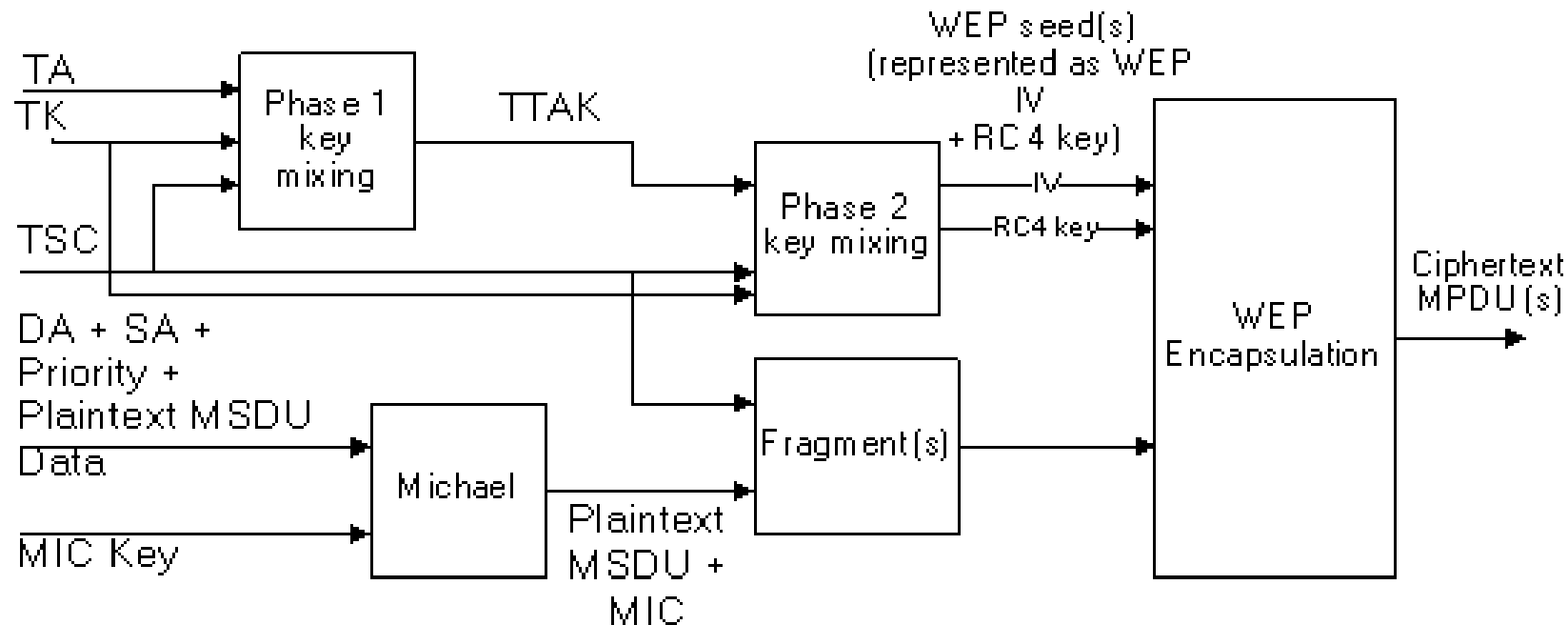


- **TTAK** TKIP Mixed Address and Key
- **TK** Temporal Key
- **TSC** TKIP Sequence Counter

- Phase 2 Key Mixing

- TTKAK = Phase1(TA, TK, TSC)
- Phase2(TTKAK, TK, TSC)
- Phase2 ist Feistel-Chiffre:
- Einfache Operationen für „schwache“ AP-Hardware
  - XOR, UND, ODER, >>
  - S-Box
- Erzeugt 128 Bit WEP-Schlüssel
  - 24 Bit Initialisierungsvektor
  - 104 Bit RC4-Schlüssel

## ■ Aus IEEE 802.1i-2004



- Für jedes Frame (MSDU) wird eigener Schlüssel generiert
- Hardware-Abwärtskompatibilität; d.h. Verwendung von RC4 nach wie vor problematisch

- Bei Verwendung von Pre Shared Keys (PSK) hängt die Sicherheit stark von der Stärke des Passworts ab
- Angriff mit Rainbow-Tables (seit 2004)
- Angriff auf PRF Funktion der Schlüsselverteilung (August 2008)
  - nutzt GPUs (Graphics Processing Units) anstatt CPUs
  - Entwickelt auf NVIDIA-CUDA (Compute Unified Device Architecture)
    - Compiler und Entwicklungsumgebung
    - nativer Zugriff auf GPUs auf Grafikkarten
    - dadurch massive Parallelisierung möglich
    - damit Speedup von Faktor 30 und mehr möglich
    - Zeit für „Raten“ eines Passwortes reduziert sich auf 2-3 Tage
- Angriff auf TKIP Verschlüsselung (November 2008)
  - Entschlüsselung von Paketen mit teilweise bekanntem Inhalt ohne Kenntnis des Schlüssels möglich
  - Schlüssel ist damit nicht zu brechen

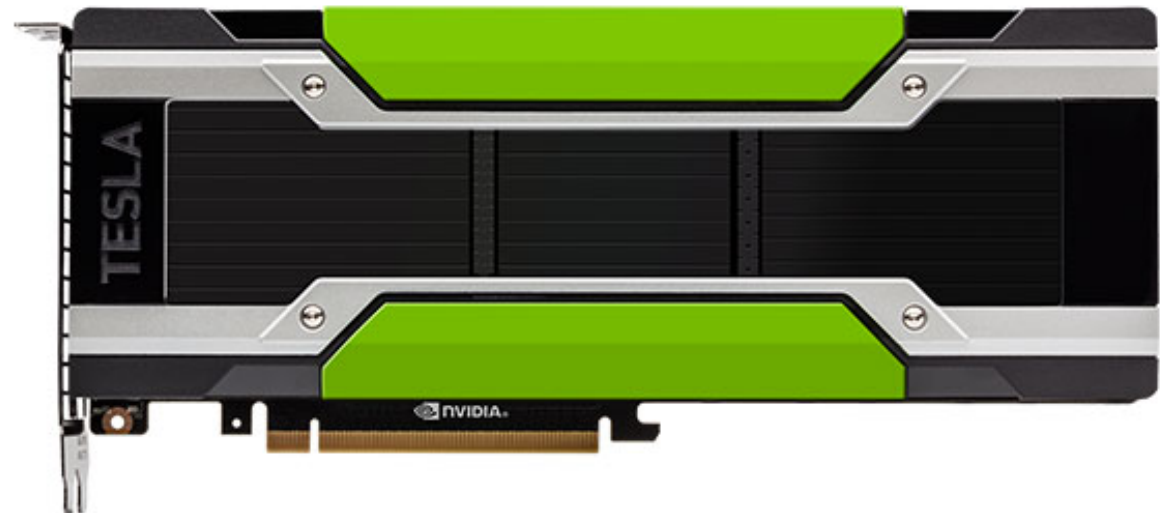
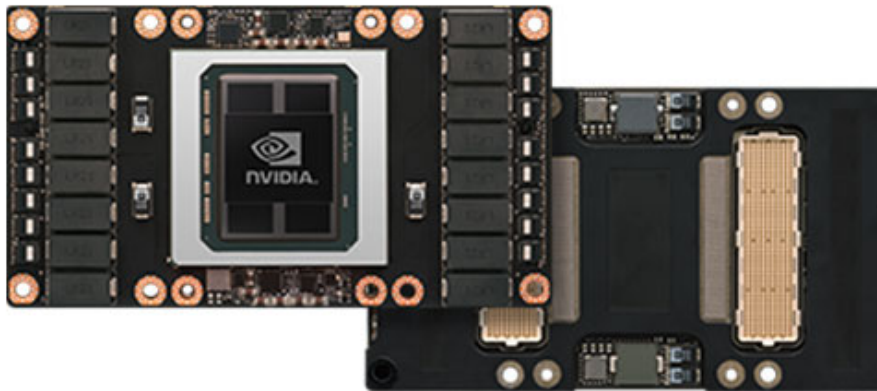
- 2007/2008
- 128 stream processors
- 330 GFlops ( $10^9$ )  
(today's general purpose CPUs have ~10)
- 150W
- Top of the line graphics hardware (along with the G92)

damals



# NVIDIA Tesla P100

- 2018: „GPU Accelerator“ for HPC Data Centers
- 3584 Cores
- 18,7 TFlops ( $10^{12}$ )  
(Intel Skylake 8168: 1,5 TFlops)
- 250 W

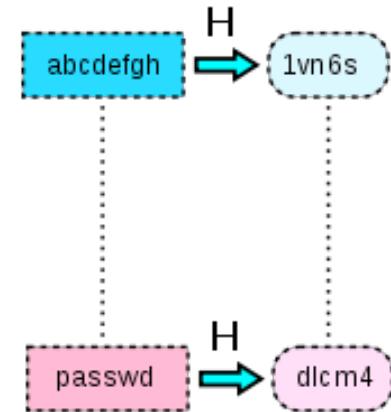




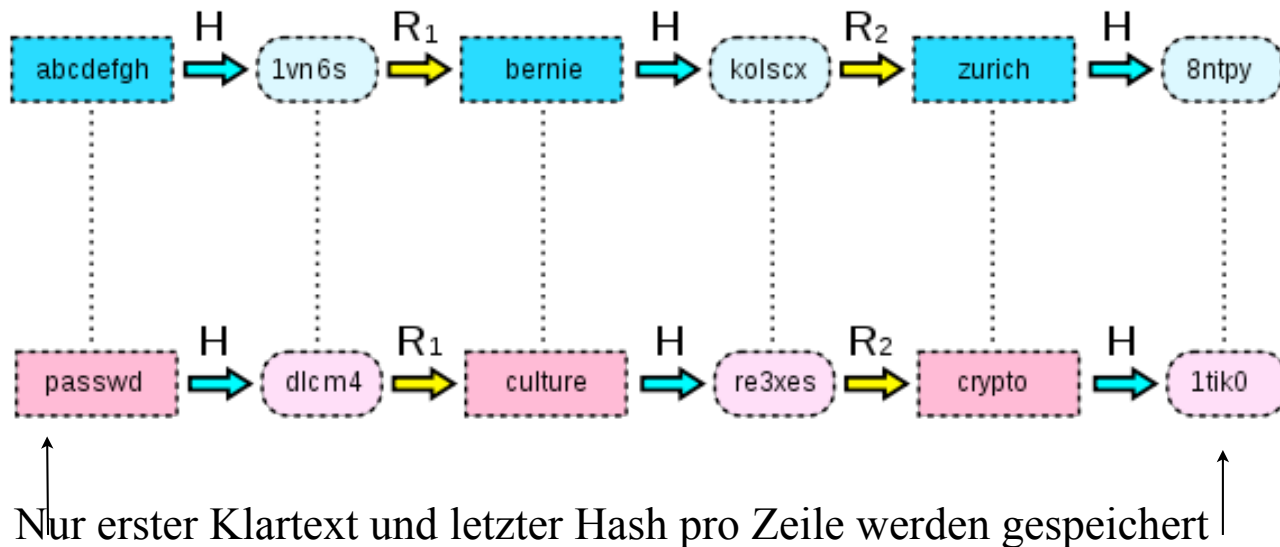
- Bei allen Krypto-Angriffen ist Rechenzeit- und Speicherplatzkomplexität zu betrachten
- Rainbow-Tables versuchen, optimalen time-memory tradeoff zu nutzen, um vollständigen Brute-Force-Angriff zu sparen
- Idee: Optimale Speicherung einer Klartext-zu-Hash Tabelle
- Kompakte Speicherung von sog. Chains (Ketten/PW-Sequenzen)
  - Kette startet mit initialem Klartext-Wort, dieses wird gehasht
  - resultierender Hash wird Reduktionsfunktion unterworfen
  - Reduktionsfunktion liefert weiteres potentiell Klartext-Wort
  - Dieser Vorgang wird n-mal wiederholt
  - relevant sind nur erstes Klartext-Wort und letzter Hash-Wert
  - Vorgang wird einmal für alle Wörter eines Wörterbuchs wiederholt
  - Kollisionen vermeiden: internes Klartext-Wort darf nicht Startwert einer anderen Kette sein

## ■ Trivialfall: Nur 1 Iteration

- Speichert zu jedem Klartext seine Hashsumme
- Rainbow-Tabelle wird sehr lang und damit zu groß



## ■ 3 Iterationen:



- Rainbow-Tabelle mit  $w$  Einträgen und Ketten der Länge  $n$
- MD5 Hash: `bca6a2aed3edc8e22f68ed65e39682c6` („IT-Sec“)
- Suche in Tabelle auf rechter Seite. Fallunterscheidung:
  1. Hash-Wert gefunden, steht z.B. in Zeile 17
    - Kette aus Zeile 17 komplett durchlaufen
    - $(n-1)$ te Anwendung der Reduktionsfunktion liefert den gesuchten Klartext
  2. Hash-Wert steht nicht in Rainbow-Table
    - Reduktion des Hashes (vereinfachtes Bsp. erste 6 Zeichen): `bca6a2`
    - `MD5(bca6a2)` liefert `3c41c8c8c5d27647d3f64937a801c90a`
    - Suche diesen Hash in Tabelle
- In der Praxis werden verschiedene Reduktionsfunktionen kombiniert
  - Ziel: Kollisionen / Wiederholungen vermeiden, um möglichst viele Klartexte abzudecken

- Beck, TU Dresden, Tews, TU Darmstadt; publ. 08.11.2008
- Erstes Verfahren, das keine Pre Shared Keys voraussetzt
- Basiert auf chop-chop Angriff (bekannt seit 2005)
- Funktionsweise:
  - Angreifer schneidet Verkehr mit, bis er verschlüsseltes ARP-Paket findet (vgl. Folien „Breaking WEP in less than 60 seconds“)
  - letztes Byte wird entfernt
  - Annahme: Byte war 0; mit XOR-Verknüpfung mit bestimmten Wert wird versucht, eine gültige Checksumme zu erzeugen
  - Paket wird an STA gesendet:
    - Inkorrekt: Paket wird verworfen
    - Korrekt: Client erzeugt MIC Failure Report Frame; Angreifer muss dann vor nächstem Versuch 60 Sekunden warten, sonst erzwungener Verbindungsabbau
  - Worst Case: 256 Tests für 1 Byte erforderlich. Praktisch: In 12 Minuten mindestens 12 Byte entschlüsselbar.

- **Sicherheitsmaßnahmen von WPA**
  - Anti-chopchop: zwei falsche MICs in 1 Minute ⇨ Verbindungsabbau
  - TSC (Sequenznummer) verhindert Wiedereinspielen
- **Gegenmaßnahmen:**
  - 60 Sekunden warten (vgl. Folie vorher)
  - Replay nicht an verwendeten, sondern an anderen Sendekanal
- **Entschlüsselung des ARP Pakets ermöglicht:**
  - Schlüsselstrom vom AP zu STA und MIC Code können ermittelt werden
  - Eigene verschlüsselte Pakete können an STA gesendet werden; z.B. zum Manipulieren von ARP-Paketeten
- **Grenzen des Angriffs**
  - Rekeying-Intervall muss ausreichend groß sein
  - QoS muss aktiviert sein, sonst stehen keine 8 Kanäle zur Verfügung
  - nur eine Richtung: AP zu STA

# WPA-Schlüssel in der Cloud knacken (12.01.2011)

---

- Angriff auf WPA-Schlüssel (Pre-Shared Keys) über die Elastic Compute Cloud (EC2) Infrastruktur von Amazon
- Prinzipiell nichts Neues, nutzt nun aber die Cluster GPU Instances
- Wörterbuch-Angriff mit 70 Millionen Wörtern; pro Amazon-Maschine rund 50.000 Wörter pro Sekunde
- Alternative z.B. [www.wpacracker.com](http://www.wpacracker.com): \$17 für Wörterbuch-Angriff mit mehr als 250 Millionen Wörtern auf 400 „herkömmlichen“ Amazon CPU Instances
- Details: <http://stacksmashing.net/2011/01/12/upcoming-black-hat-talk/>

- Empfehlung: Verwendung von WPA 2 anstelle von WPA
  
- Änderungen:
  - AES ersetzt verpflichtend RC4
  - CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) als Ersatz für TKIP
  
- Verfahren gilt derzeit als sicher
  - Verpflichtend für Geräte mit Wi-Fi Logo
  
- Aber: Verschlüsselung schützt nicht ewig
  - Mitgehörte Daten können evtl. später entschlüsselt werden