HUMBOLDT-UNIVERSITÄT ZU BERLIN

Wireless Security

IT Security Workshop 2006

Moritz Grauel
grauel@informatik.hu-berlin.de
Matthias Naber
naber@informatik.hu-berlin.de

HU-Berlin - Institut für Informatik

29.09.2006

Überblick

1 WEP - Wireless Equivalent Privacy?

Überblick

1. WEP - Wireless Equivalent Privacy?
2. WEP - Schwachstellen

Überblick

1. WEP - Wireless Equivalent Privacy?
2. WEP - Schwachstellen
3. Unser Ansatz

Überblick

1. WEP - Wireless Equivalent Privacy?
2. WEP - Schwachstellen
3. Unser Ansatz
4. WPA

Überblick

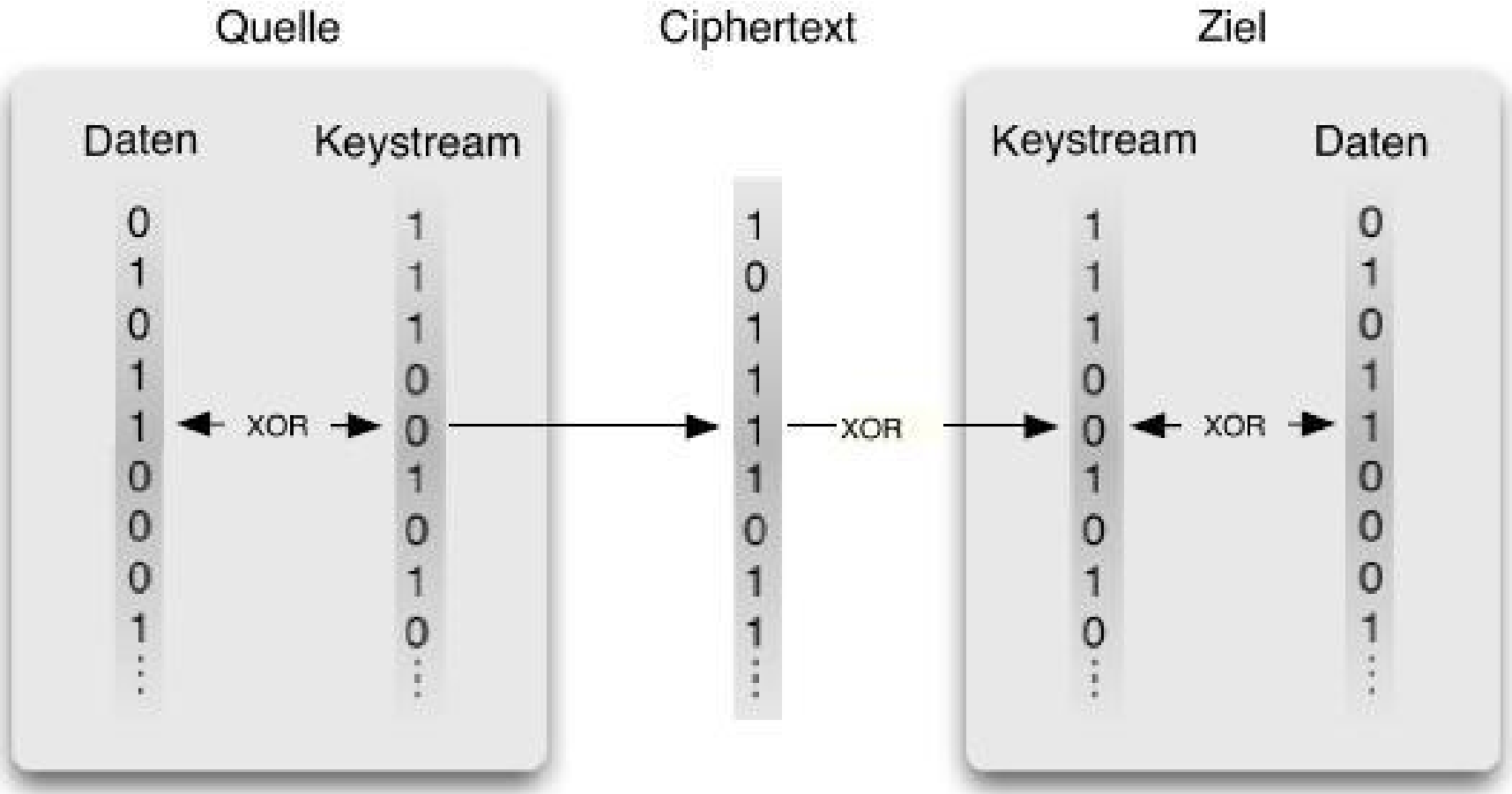
1. WEP - Wireless Equivalent Privacy?
2. WEP - Schwachstellen
3. Unser Ansatz
4. WPA

WEP - Funktionsweise

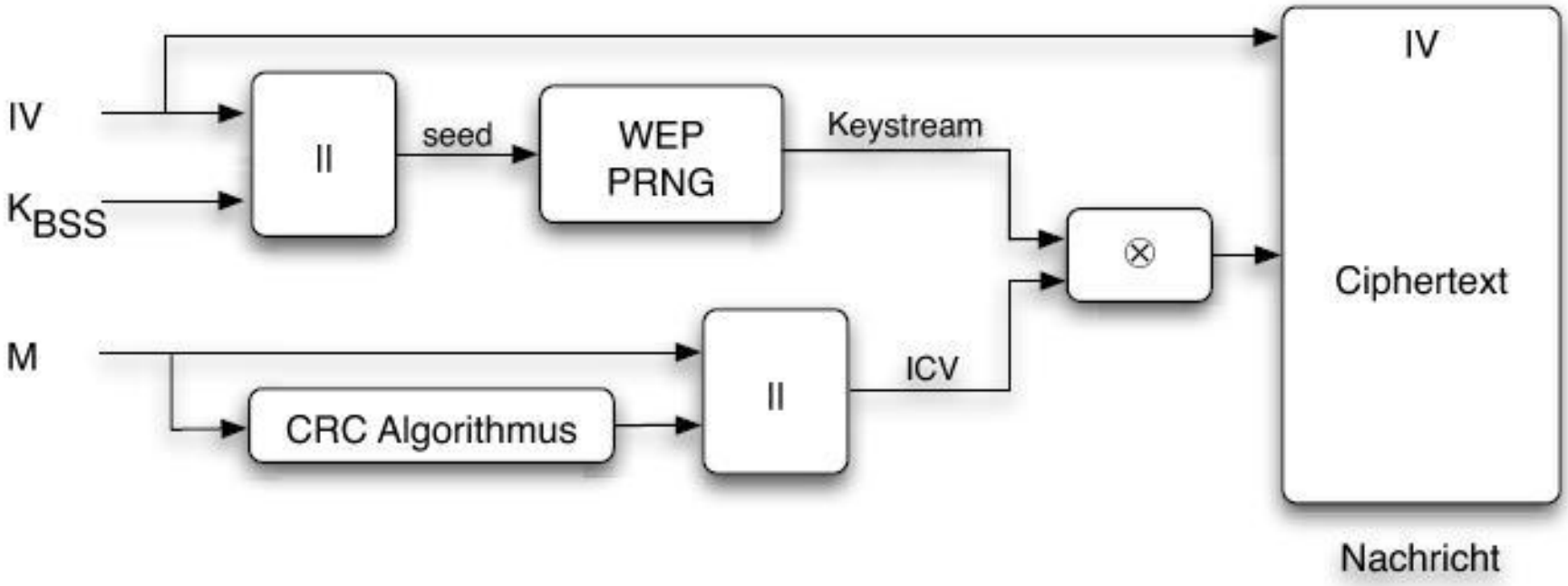
Eigenschaften

- Verschlüsselung auf Layer 3
- symmetrischer Stromchiffre
- RC4-Pseudozufallszahlengenerator
- offizielle Schlüssellängen 40 (64) Bit und 104 (128) Bit

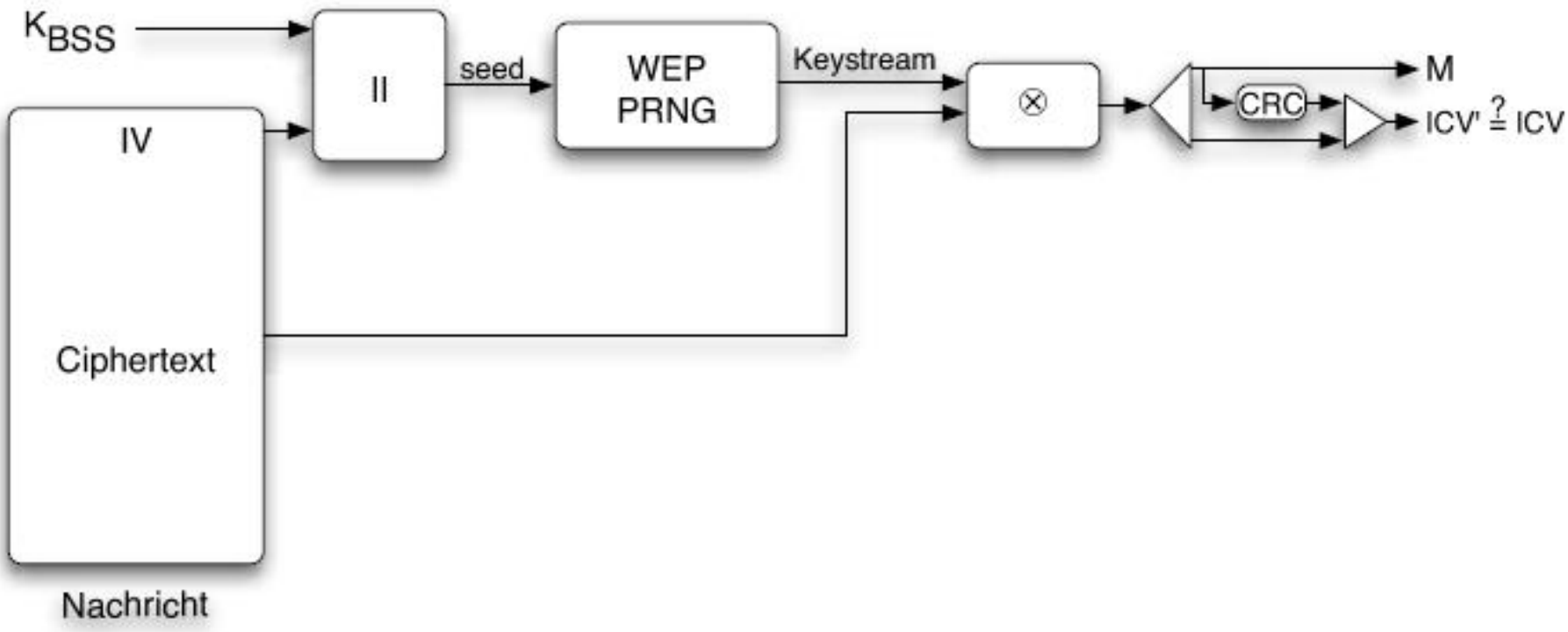
WEP-Funktionsweise - Stromchiffre



WEP-Funktionsweise - RC4



WEP-Funktionsweise - RC4



WEP-Funktionsweise - Paketaufbau

24 Byte	4 Byte	1 - 2308 Byte	4 Byte	4Byte
802.11 Header	IV Header	Frame Body	ICV	FCS
Klartext	Klartext	Verschlüsselt	Verschlüsselt	Klartext

Überblick

1 WEP - Wireless Equivalent Privacy?

2 WEP - Schwachstellen

3 Unser Ansatz

4 WPA

WEP-Schwachstellen

40 Bit Schlüssel?!

WEP-Schwachstellen

Keystream-Wiederverwendung

- das Schlimmste, das man bei Stromchiffren machen kann
- $\exists 2^{24}$ Initialisierungsvektoren
- viele AP fangen bei 0 an und inkrementieren dann

$$
\begin{aligned}
C_1 &= P_1 \oplus RC4(iv \| key) \\
C_2 &= P_2 \oplus RC4(iv \| key) \\
C_1 \oplus C_2 &= (P_1 \oplus RC4(iv \| key)) \oplus (P_2 \oplus RC4(iv \| key)) \\
\Rightarrow &= P_1 \oplus P_2
\end{aligned}
$$

WEP-Schwachstellen

KoreK

- Schwachstelle in RC4
- Anfang des Schlüsselstreams hängt von wenigen Bytes des IV ab (schwache IV)
- mindestens die ersten 8 Bytes des *Frame Bodies* sind bekannt
- ab ca 150.000 bis 300.000 unterschiedlichen IV ist der WEP-Schlüssel zurück zu rechnen
- $\Rightarrow$ aircrack

WEP-Schwachstellen

Standard-aircrack-Angriff

1. Fake-Authentisierung am AP
2. Warten auf ARP-Request (zu erkennen an fester Größe)
3. ARP-Request erneut an den AP senden
4. AP antwortet (mit neuem IV)

3. und 4. so lange Wiederholen, bis genug IVs für KoreK gesammelt sind.

WEP-Schwachstellen

Standard-aircrack-Angriff

1. Fake-Authentisierung am AP
2. Warten auf ARP-Request (zu erkennen an fester Größe)
3. ARP-Request erneut an den AP senden
4. AP antwortet (mit neuem IV)

3. und 4. so lange Wiederholen, bis genug IVs für KoreK gesammelt sind. Dauert ungefähr 4 bis 6 Stunden.

WEP-Schwachstellen

Schwache Passwörter

- ASCII Passwörter
- Windows verlangt 5 oder 13 Zeichen
- Schlüsselraum 256^5 bei Hex
- Schlüsselraum 62^5 bei ASCII

WEP-Schwachstellen

Integritätsprüfung

- Hashfunktion ist CRC32

WEP-Schwachstellen

Integritätsprüfung

- Hashfunktion ist CRC32
- CRC32 ist kein kryptographischer Hash
- bedingt vorhersehbar (Zip-Recovery)
- gezielte Manipulation ist möglich
- $\Rightarrow$ chop-chop

WEP-Schwachstellen

Fragmentierung

- Fragmentierung auf MAC-Ebene ist zulässig
- maximal 16 Fragmente pro Paket
- Fragmente dürfen mit gleichem IV verschlüsselt sein

Überblick

1 WEP - Wireless Equivalent Privacy?

2 WEP - Schwachstellen

3 Unser Ansatz

4 WPA

Unser Ansatz

Hardware

- Opfer: Netgear WGT634U *out-of-the-box*

Unser Ansatz

Hardware

- Opfer: Netgear WGT634U *out-of-the-box*
- Angreifer: Netgear WGT634U mit gepatchter Software
- OpenWGT-Linux
- Atheros-WLAN-Chip

Unser Angriff

Fragmentierung

- WEP-Paket

24 Byte	4 Byte	1 - 2308 Byte	4 Byte	4Byte
802.11 Header	IV Header	Frame Body	ICV	FCS
Klartext	Klartext	Verschlüsselt	Verschlüsselt	Klartext

- Frame Body eines ARP-Requests

LLC/SNAP	ARP Header	Quelle	Ziel
AA AA 03 00 00 00 08 06			

- LLC/SNAP ist fast immer Konstant
- Wir kennen den IV und 8 Byte des Schlüsselstroms

Unser Angriff

IV KeyIdx	LLC/SNAP	ARP Header	Q	Z
40 56 B6 00	80 D1 F3 FB EA 6D 46 67	...	...	...
	$\oplus$			
Klartext	AA AA 03 00 00 00 08 06	...	...	...
	$\equiv$			
Schlüssel	2A 7B F0 FB EA 6D 4E 61	...	...	...

Unser Angriff

Daten Fragmentieren

- erstellen eines ARP-Requests (36 Byte)
- fragmentieren in 4 Byte-Fragmente (4 Byte Daten + 4 Byte ICV)
- Fragmente verschlüsseln, in 9 Pakete verteilen und versenden
- Antwort abwarten

Unser Angriff

Daten Fragmentieren

- erstellen eines ARP-Requests (36 Byte)
- fragmentieren in 4 Byte-Fragmente (4 Byte Daten + 4 Byte ICV)
- Fragmente verschlüsseln, in 9 Pakete verteilen und versenden
- Antwort abwarten
- Wir kennen nun einen (neuen) IV und 40 Byte Schlüsselstrom

Unser Angriff

Was haben wir jetzt gekonnt?

- Wir könnten beliebige Pakete mit bis zu 576 Byte Länge verschlüsseln und schicken (16 * 36 Byte)
- Wir könnten uns noch längere Schlüsselströme erzeugen
- Wir könnten uns genug Traffic erzeugen

Unser Angriff

Probleme

- „Endianism“
- Speicherplatz und NFS
- Cross-Compiler
- Fake-Authentifizierung (aircrack)
- madwifi-Treiber
- Sequenznummer
- Checksummen
- Antwort/Forward
- unbekannter SNAP 0x0027
- viel Noise

Unser Angriff

Was haben wir erreicht?

- wir können beliebige fragmentierte Pakete an die Broadcast-Addresse schicken
- der AP setzt sie zusammen und verschickt den Broadcast
- wir haben knapp 1000 Byte Schlüsselstrom (verifiziert!)

Überblick

1. WEP - Wireless Equivalent Privacy?
2. WEP - Schwachstellen
3. Unser Ansatz
4. WPA

WPA

Verschiedene Arten von WPA

- WPA
- WPA2
- WPA-EAP
- WPA-PSK
- 802.11i

WPA

Übersicht

- Erweiterung von WEP
- benutzt somit auch RC4
- dynamische Schlüsselgenerierung durch *Temporary Key Integrity Protocol (TKIP)*
- Authentizierung durch *PreSharedKey (PSK)* oder *Extensible Authentication Protocol (EAP)*
- IV = 16Bit-Lo $\|$ 32Bit-Hi
- *Message Integrity Check (MIC oder Michael)*
- Schlüssellänge 63 ASCII-Zeichen oder 128Bit

WPA2

Erweiterungen zu WPA

- nutzt EAS anstelle von RC4
- im Standard 802.11i enthalten

Advanced Encryption Standard

Iterierte Blockchiffre

R	128	192	256
128	10	12	14
192	12	12	14
256	14	14	14

Advanced Encryption Standard

Rundenfunktion

ADDROUNDKEY (K_0)
for i:= 1 downto 9 do
SUBBYTES (Substitutions-Box)
SHIFTROWS (Permutation)
MIXCOLUMNS (lineare Substitution)
ADDROUNDKEY(K_i)
end
SUBBYTES
SHIFTROWS
ADDROUNDKEY(K^10)

Ende. . .

Noch fragen?

Quellen

- Paper von Andrea Bittau
  `http://tapir.cs.ucl.ac.uk/bittau-wep.pdf`
- Paper von Matthias Valentin `valentin@net.in.tum.de`
- 802.11 Wireless Networks - O'Reilly Verlag
- CRC-Implementation vom SAR-Lehrstuhl
- RC4-Implementation
  `http://groups.google.com/group/sci.crypt/msg/10a300`
- Wikipedia
- Google ;-)