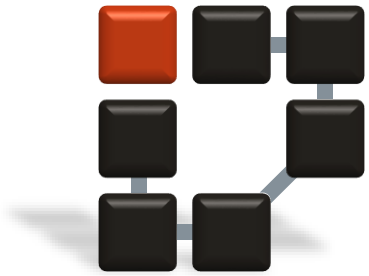
Wired Equivalent Privacy (WEP)

Dr.-Ing. Abdalkarim Awad
27.1.2016



Informatik 7
Rechnernetze und
Kommunikationssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

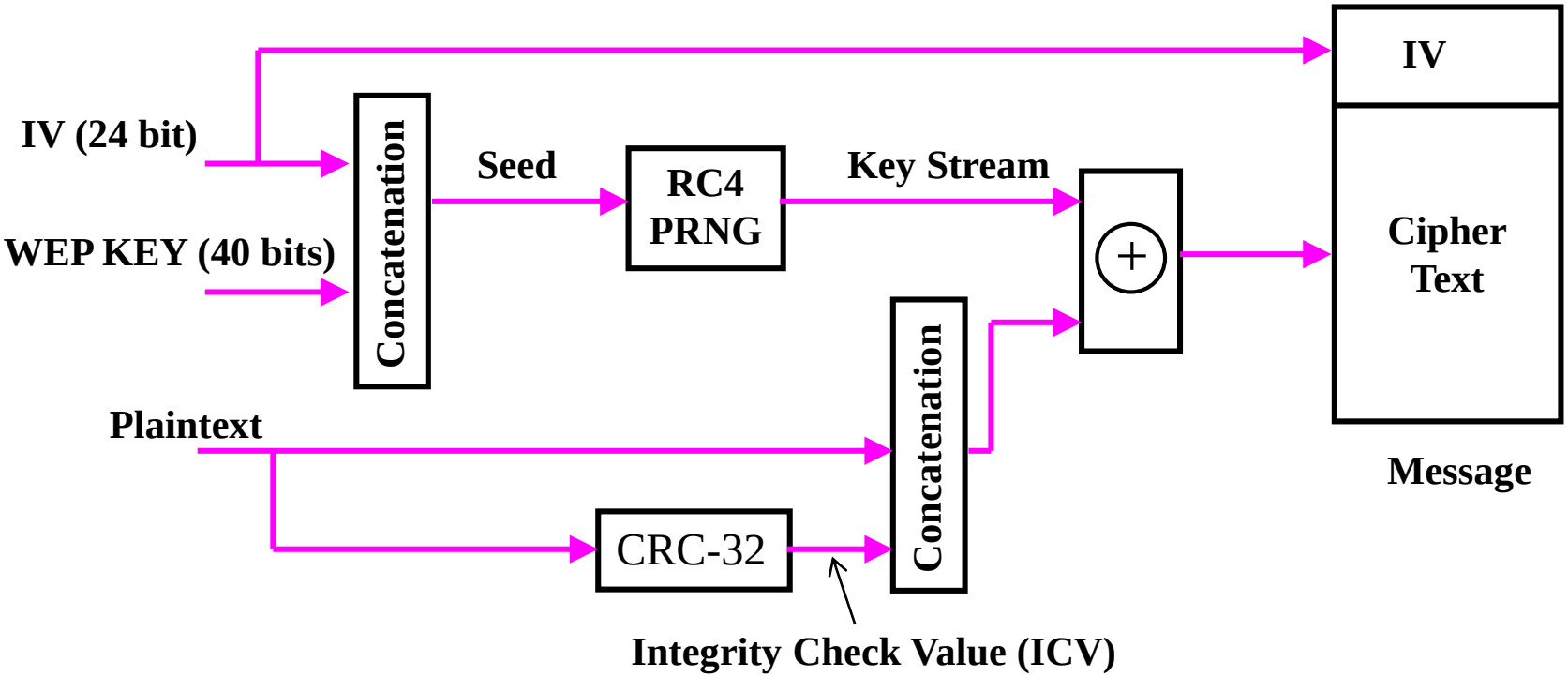
TECHNISCHE FAKULTÄT

Ethical issues

- The goal of this exercise to learn about the weakness of WEP.
- It is not intended to be used as a tool to steal information damage systems.

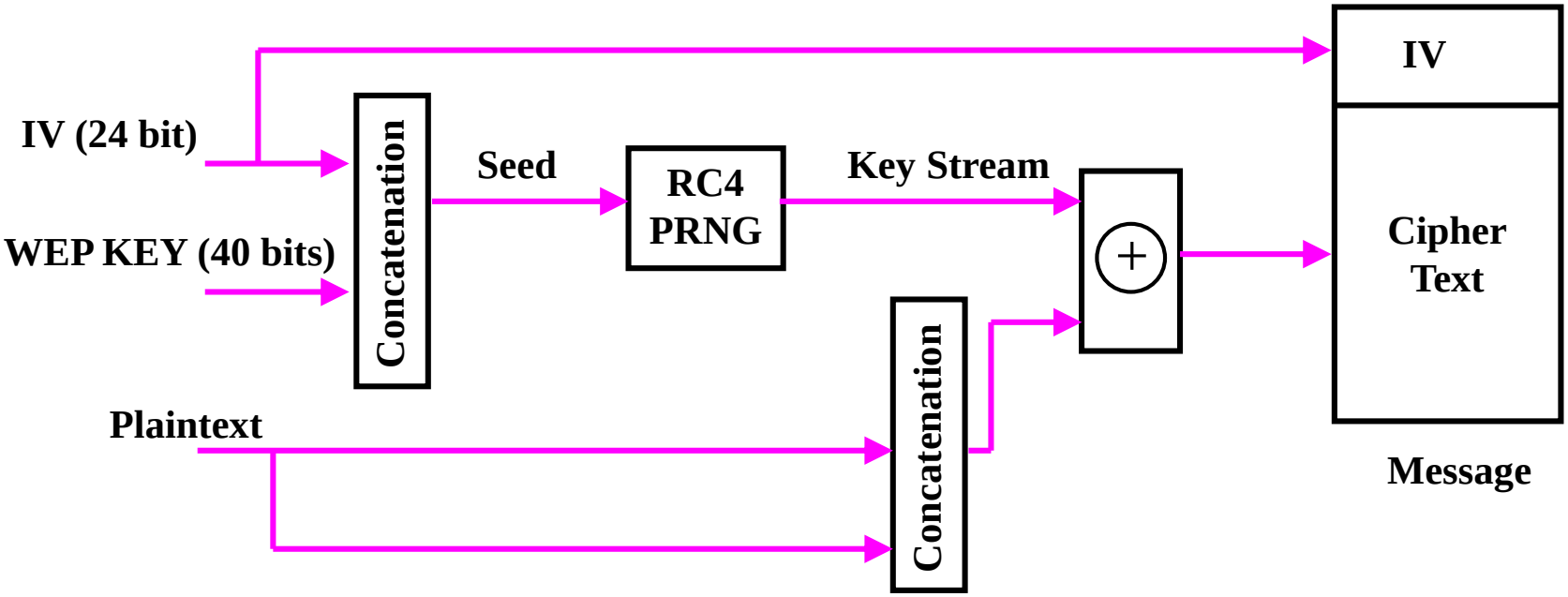
Wired Equivalent Privacy (WEP)

- WEP Encryption uses RC4 stream cipher



Wired Eequivalent Privacy (WEP)

- WEP decryption uses RC4 stream cipher



RC4

- a proprietary cipher owned by RSA DSI
- another Ron Rivest design, simple but effective
- variable key size, byte-oriented stream cipher
- widely used (web SSL/TLS, wireless WEP/WPA)
- key forms random permutation of all 8-bit values
- uses that permutation to scramble input info processed a byte at a time

RC4: Two Steps

- RC4 Key Schedule (Initialization)

```
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) (mod 256);
    swap (S[i], S[j]);
```

- RC4 Encryption

```
i = j = 0;
for each message byte M_i
    i = (i + 1) (mod 256);
    j = (j + S[i]) (mod 256);
    swap(S[i], S[j]);
    t = (S[i] + S[j]) (mod 256);
    C_i = M_i XOR S[t];
```

RC4 Key Schedule

- starts with an array S of numbers: 0..255
- use key to well and truly shuffle
- S forms **internal state** of the cipher

```
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) (mod 256);
    swap (S[i], S[j]);
```

RC4 Encryption

- encryption continues shuffling array values
- sum of shuffled pair selects "stream key" value from permutation
- XOR S[t] with next byte of message to en/decrypt

```
i = j = 0;
for each message byte M_i
    i = (i + 1) (mod 256);
    j = (j + S[i]) (mod 256);
    swap(S[i], S[j]);
    t = (S[i] + S[j]) (mod 256);
    C_i = M_i XOR S[t];
```

WEP decryption step-by-step

Step 1: Build the keystream

- Extract the IV from the incoming frame
- Prepend the IV to the key
- Use RC4 to build the keystream

WEP decryption step-by-step

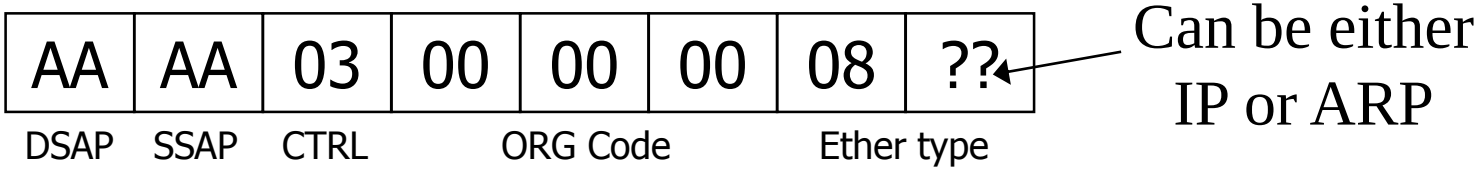
Step 2: Decrypt the plaintext and verify

- XOR the keystream with the ciphertext
- Verify the extracted message with the some known data in the packet

Initialization vector (IV)

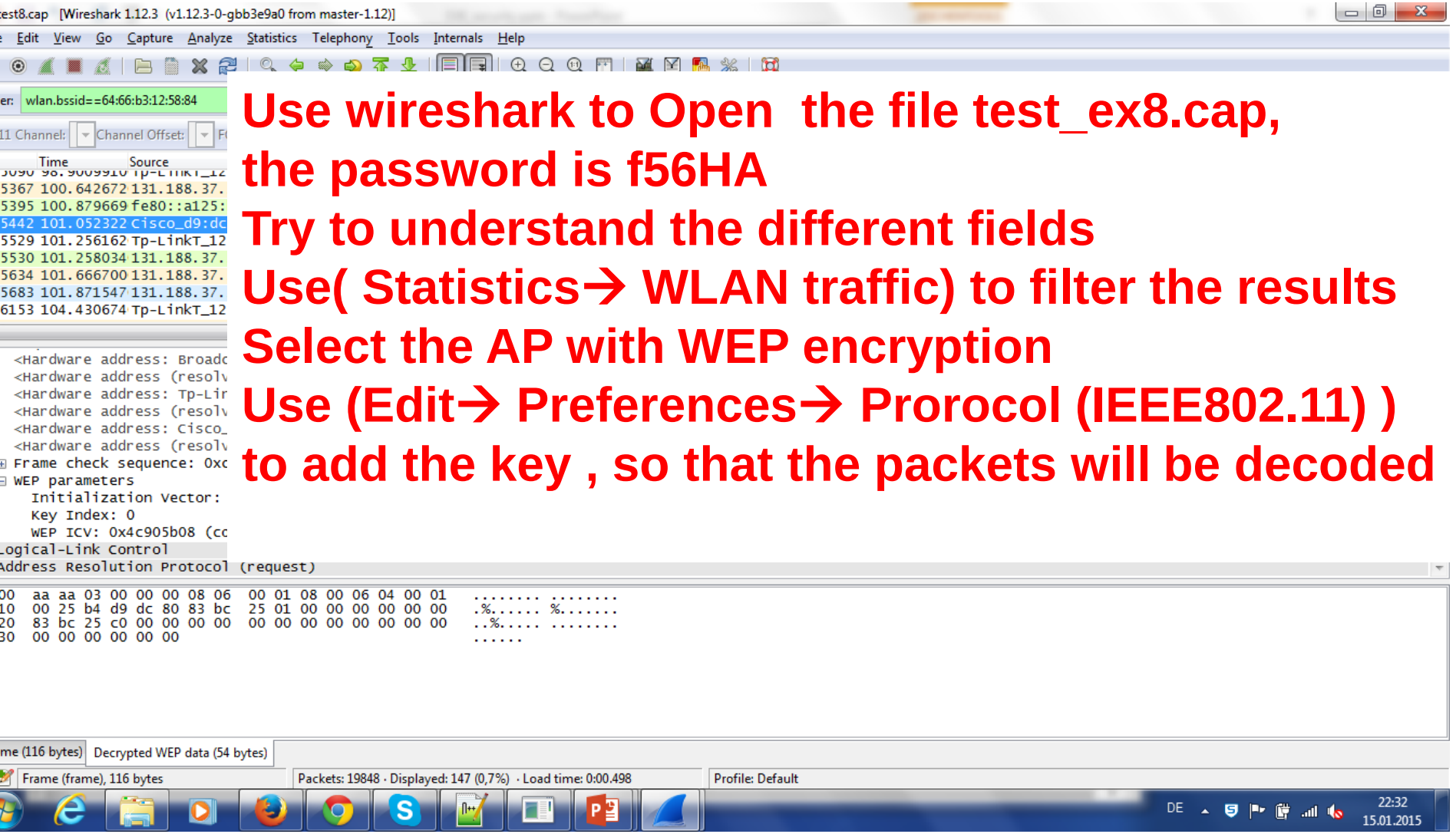
- It's carried in plaintext in the "encrypted" message!
- It's only 24 bits!
- There are no restrictions on IV reuse!
- The IV forms a significant portion of the "seed" for the RC4 algorithm!

What we know about the packets



- With 802.11, you know the first eight bytes of a packet
- Many IP services have packets of fixed lengths
- Most WLAN IP addresses follow common conventions.
- Many IP behaviors have predictable responses
- The network part of IP address is known

Example



Use wireshark to Open the file test_ex8.cap, the password is f56HA

Try to understand the different fields

Use(Statistics→ WLAN traffic) to filter the results

Select the AP with WEP encryption

Use (Edit→ Preferences→ Prorocol (IEEE802.11)) to add the key , so that the packets will be decoded

ARP packet

15442	101.052322	Cisco_d9:dc:80	Broadcast	ARP	116 Who has 131.188
15529	101.256162	Tp-LinkT_12:58:84	Broadcast	802.11	140 Beacon frame, S
15530	101.258034	131.188.37.28	239.255.255.250	SSDP	231 M-SEARCH * HTTP
15634	101.666700	131.188.37.28	224.0.0.2	IGMPv2	116 Leave Group 224

```
    <Hardware address: Cisco_d9:dc:80 (00:25:b4:d9:dc:80)>
    <Hardware address (resolved): Cisco_d9:dc:80>
  ⊞ Frame check sequence: 0xcf014bc3 [correct]
  ⊟ WEP parameters
      Initialization Vector: 0x778e26
      Key Index: 0
      WEP ICV: 0x4c905b08 (correct)
⊞ Logical-Link Control
⊞ Address Resolution Protocol (request)

0000  aa aa 03 00 00 00 08 06  00 01 08 00 06 04 00 01   ........ ........
0010  00 25 b4 d9 dc 80 83 bc  25 01 00 00 00 00 00 00   .%...... %.......
0020  83 bc 25 c0 00 00 00 00  00 00 00 00 00 00 00 00   ..%..... ........
0030  00 00 00 00 00 00                                 ......
```

IV

Some common data in all ARP packets

Cracking the password

- Brute Force method
- Get the IV from an ARP packet (data packet)
- Get the encrypted data from the Packet as hex
- Assume the password consists from small/capital letters in addition to numbers
- Concatenate a 40 bits (5 chars) key to have the complete Key.
- Key schedule, obtain the vector S based on the key
- Using the encrypted data and S, decode the encrypted message and compare the results in byte 0, 1, 2,3,and 4, with 0xaa, 0xaa, 0x00, 0x00, 0x00.
- If the results are true, then the password is cracked

Bibliography

- Smart Grid: Technology and Applications, 2012, ISBN 1119968682, Wiley, by Janaka Ekanayake, Kithsiri Liyanage, Jianzhong Wu, Akihiko Yokoyama, Nick Jenkins
- Smart Grid : Applications, Communications, and Security by Lars T. Berger and Krzysztof Iniewski
- Computer Networks A Top-Down Approach, James F. Kurose and Keith W. Ross
- Computer Networks A Top-Down Approach (Slides)
- Cryptography and Network Security, William Stallings
- Cryptography and Network Security Lecture slides by Lawrie Brown
- Security and Cryptography, Steven Gordon