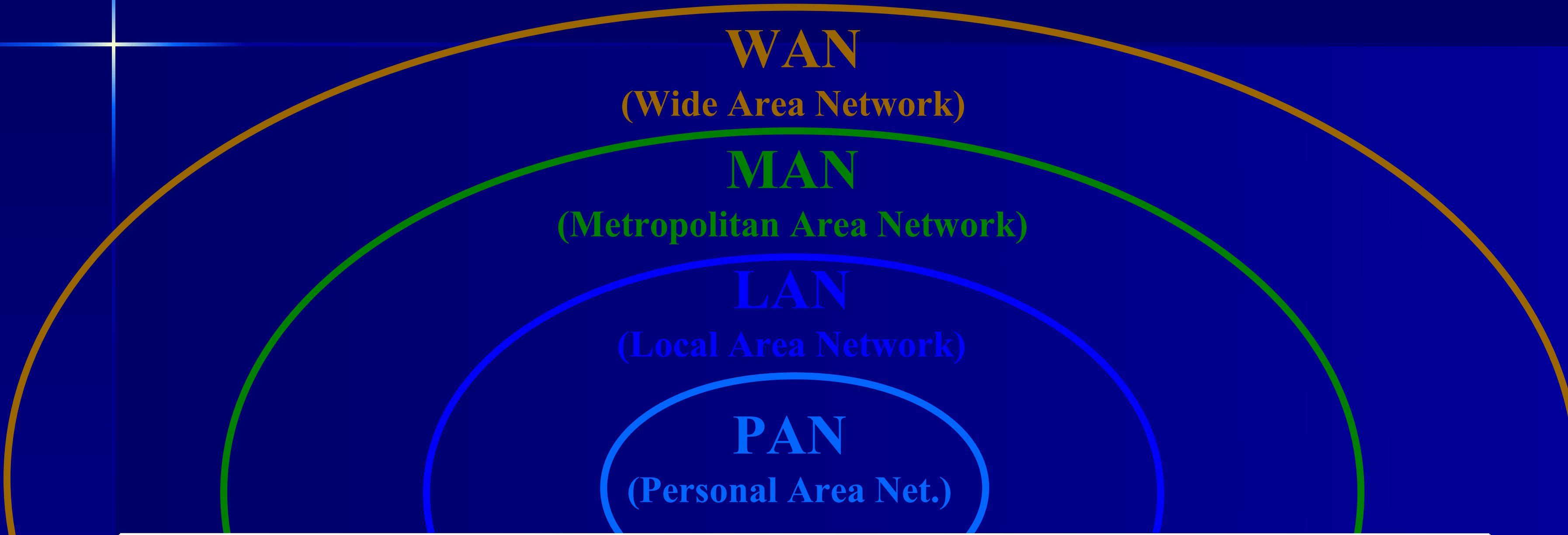
Security of Wireless LANs (IEEE 802.11)

Stefano Zanero

Wireless Technologies



	PAN	LAN	MAN	WAN
Standard	Bluetooth	802.11 HiperLAN2	802.11 MMDS, LMDS	GSM, GPRS, CDMA, 2.5-3G
Data rate	< 1Mbps	11 to 54 Mbps	11 to 100+ Mbps	10 to 384Kbps
Raggio	Short	Medium	Medium-Long	Long
Applicazioni	Peer-to-Peer Device-to-Device	Enterprise networks	T1 replacement, last mile access	PDAs, Mobile Phones, cellular access

The wireless security problem

- Any wireless network is unsafe as a wired network PLUS the intrinsic risks related to being radio-transmitted
- We appreciate wifi because it spreads all over. The risk is that it really spreads all over...
- Anything transmitted by radio without encryption can be picked up by any receiving station in range (remember Enigma?)
- Authentication is also an issue, as wireless cannot be “physically” contained as cable access can. We literally risk placing a network plug connected to the internal network in the parking lot!
- Availability is also a problem, as radio transmitters are prone to physical denial of service attacks

Wireless LAN standards

- IEEE 802.11: 1997, 2.4GHz band, 1-2Mbps
- All 802.11 networks use the same protocols for Media Access Control (MAC), namely Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
- 802.11a: 1999, 5GHz band, OFDM encoding, up to 54Mbps
- 802.11b: 1999, 2,4 GHz band, backward-compatible, DSSS encoding, 11 Mbps
- 802.11g: 2001, 2,4 Ghz band, backward-compatible, OFDM/CCK encoding, 54Mbps

IEEE 802.11b

- 2.4 GhZ ISM band
- Uses 14 channels for phi-layer separation (actually, just 3 are totally separated) but multiple networks with a different SSID (Service Set ID) are logically separated
- Shared medium
- Two modes: infrastructure (with base stations/access points, possibile roaming e bridging) or ad-hoc
- Range approx 100m outdoor, 50m indoor

Security protocol of IEEE 802.11b: WEP

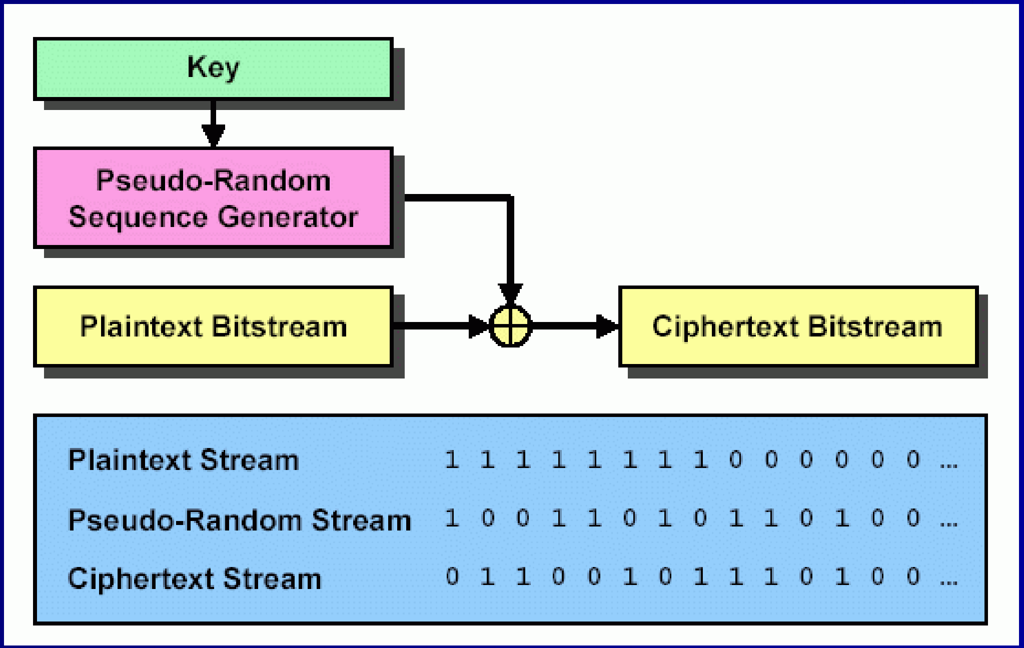
- Designing the IEEE 802.11b standard, authentication, confidentiality and integrity of data were considered, and the target was set to make them “equivalent” to the ones on a wired network
- The WEP (Wired Equivalent Privacy) protocol was designed, as we will see, with a series of design flaws

ID card of WEP

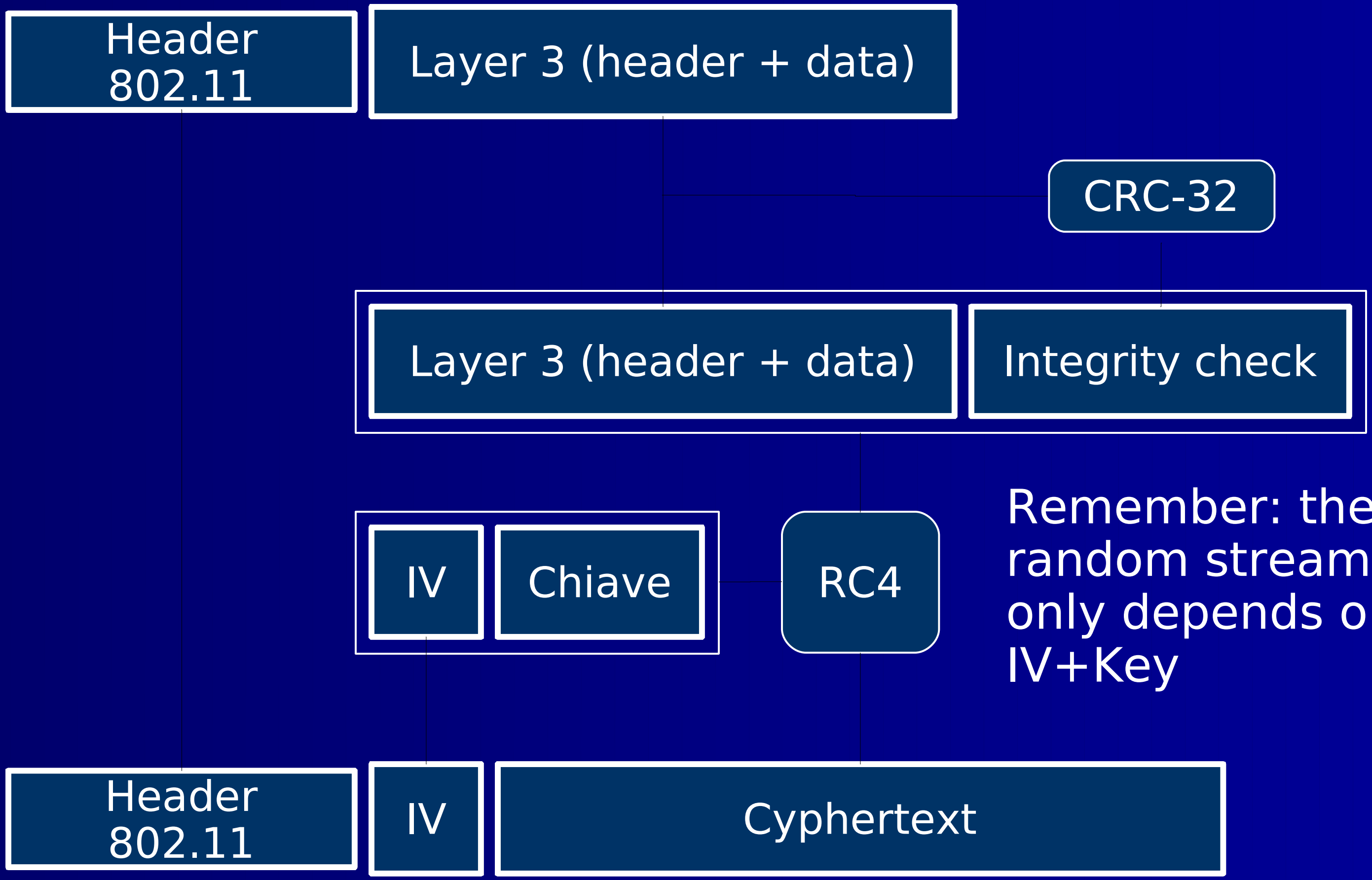
- WEP is based on a CFB stream cipher (back to this in the next slide) based on a shared key which must be manually set on all the clients and the APs
- It is based on the RC4 stream cipher algorithm designed by Ron Rivest in 1987 and protected as a trade secret by RSA until 1994, when it was leaked on USENET and became public domain
- Public domain is good for standards!

What is a CFB stream Cipher ?

- Stream cipher: bit-per-bit XOR between the plaintext stream and a pseudorandom stream
- The stream is generated by the key only
- If the key stays the same…



WEP in detail



Remember: the random stream only depends on IV+Key

WEP in detail

- Since for a SC in CFB mode:
  - Same Plain Text results in same Cipher Text
  - Key stream is always the same
- To mitigate, a random IV is used
  - combines with the key and adds randomness
  - Needs to be transmitted in clear (but this is fine, because it will combine with the key)
- Exporting ciphers above 64bits forbidden by ITAR: 128 bit version of RC4 could not be used
  - 64 bits = 24 bits IV + 40 bits key
  - Later on, they couldn't change the format, so 128 bits = 24 bits IV + 104 bits key

Born to be broken...

- RC4 could be exported only in the 40+24 bit version due to ITAR restriction, 104+24 bit version was cleared only later
- 802.11 uses CRC-32 as a MIC (Message Integrity Code). CRC-32 is distributive wrt XOR:
  CRC(A xor B) = CRC(A) xor CRC(B)
- RC4 uses XOR to encrypt...

The breakup (1)

- 2000: J. Walker studies reuse of IV in WEP
  - Space is small (2^{24})
  - Birthday paradox makes for a high probability of overlap
  - APs which use the same pseudorandom sequence obviously are a problem
  - If APs try to divide the space, situation grows worse

The breakup (2)

- 2001: Borisov, Goldberg and Wagner show practical reuse attacks
- They also describe a method to flip arbitrary bits into encrypted messages using the fact that CRC is distributive wrt XOR
  - Original = A | CRC (A)
  - Want to send A xor M, need to build CRC(A xor M), which is CRC(A) xor CRC(M)
- This depends from the lack of an authenticated portion in CRC-32

The breakup (3)

- W. Arbaugh creates a step-by-step attack to retrieve the key stream (not the key):
  - Let's suppose we know the plaintext of n bytes of ciphertext (e.g. a DHCP request, a DNS request...)
  - We know, therefore, n bytes of the key stream associated to some IV
  - We can therefore inject arbitrary message of size (n-4)
  - Pick a message long (n-3) which generates an answer if received (e.g. a ping)
  - Encrypt it and guess the last byte; if answer received we guessed right
  - lather, rinse, repeat

The breakup (4)

- Walker: IV is a bad idea per se
- Borisov-Goldberg-Wagner: no integrity, even if attacker knows nothing about key or keystream
- Arbaugh: practical recovery of keystream, building a dictionary with a cost of time (average, at date, 18h, worst case 55h) and space (several gigabytes); active attack (i.e. requires to transmit a lot of packets)
- None of these attacks directly compromises the WEP key

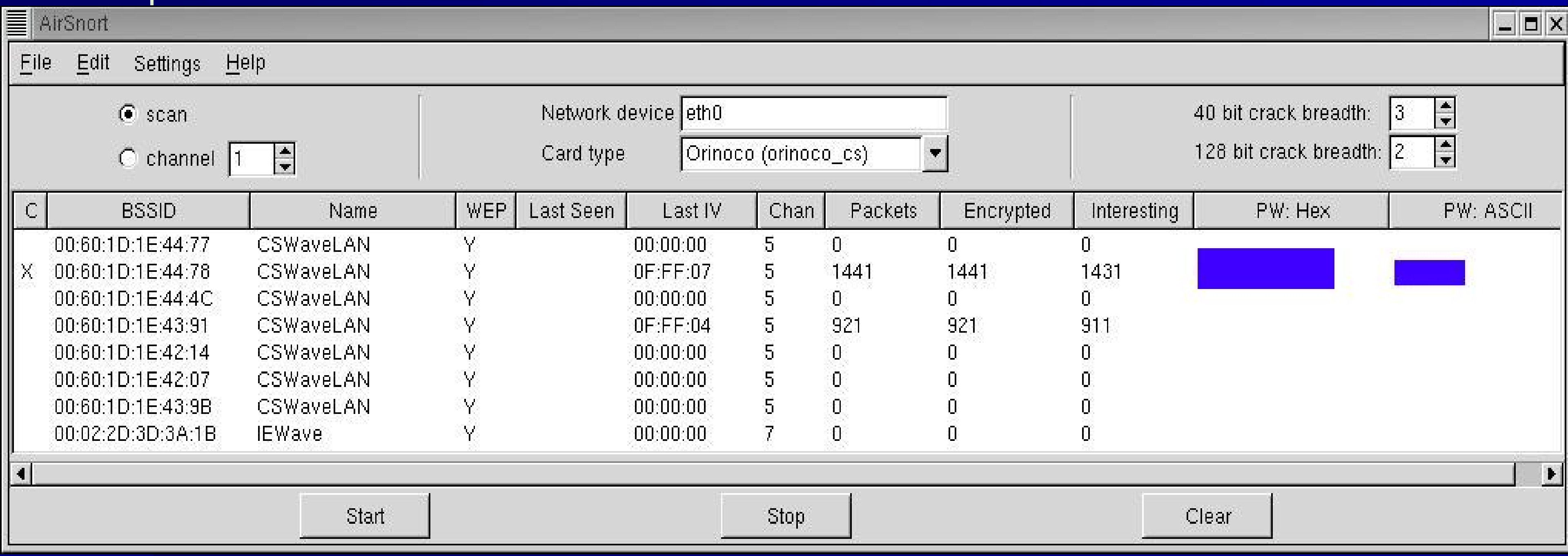
Final hit

- Fluhrer, Shamir, Mantin describe a vulnerability when RC4 is used with a fixed key part and a variable key part
- They develop a passive statistic attack which extracts information on the key directly from the ciphertext, exploiting a set of weak IV
- Stubblefield, Ioannidis and Rubin implement it against WEP
- The attack needs several million packets (several hours of sniffing) and then breaks the key in a few seconds. It is completely passive.

Tool

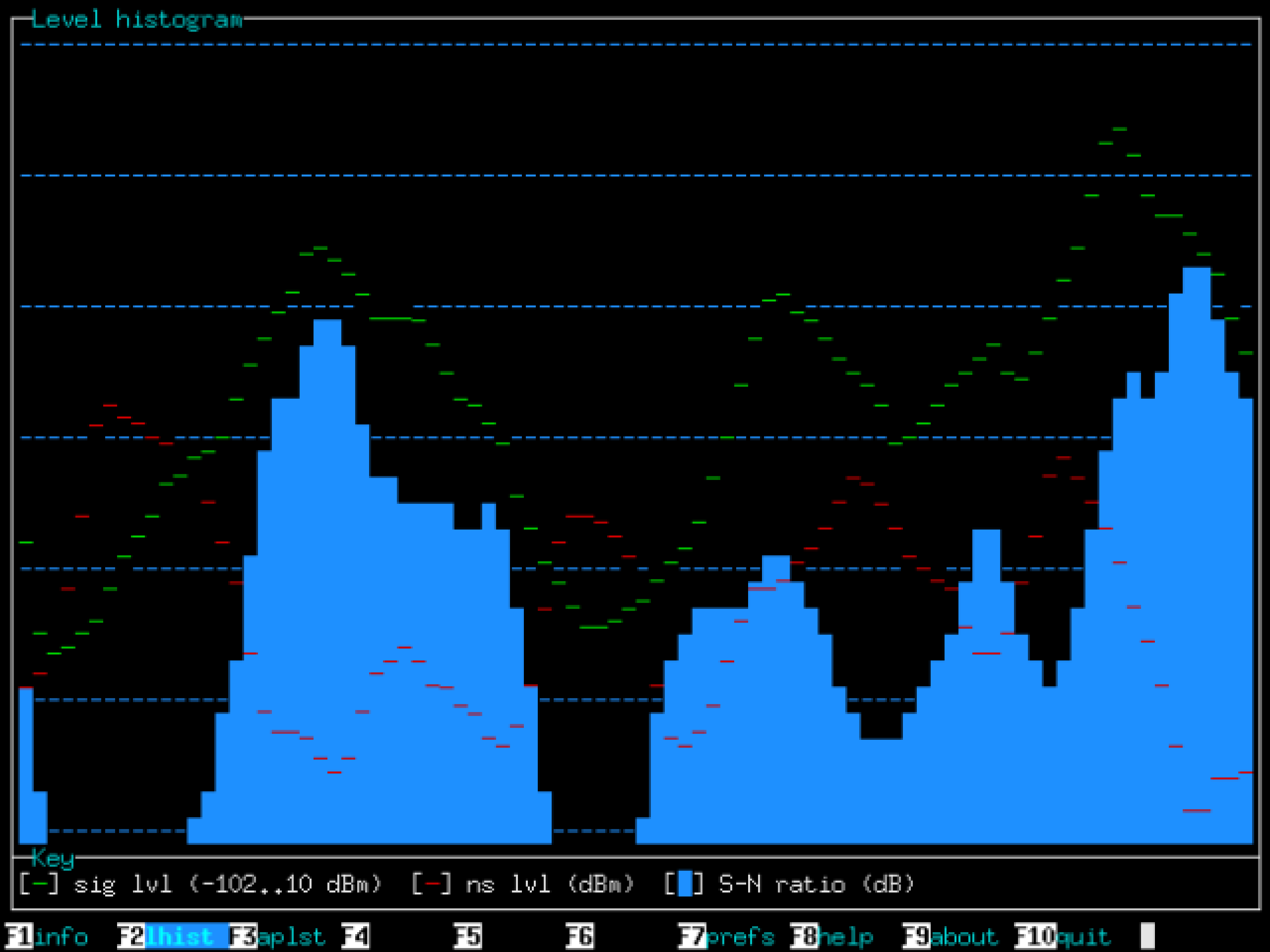
- AirSnort
  http://airsnort.shmoo.com/
- Implements the attack as described by Stubblefield et al. (but rewritten from scratch)
- Also WepCrack:
  http://sourceforge.net/projects/wepcrack

AirSnort


AirSnort
File
Edit
Settings
Help
scan
channel
1
Network device
eth0
Card type
Orinoco (orinoco_cs)
40 bit crack breadth:
3
128 bit crack breadth:
2
C
BSSID
Name
WEP
Last Seen
Last IV
Chan
Packets
Encrypted
Interesting
PW: Hex
PW: ASCII
00:60:1D:1E:44:77 CSWaveLAN Y 00:00:00 5 0 0 0
X 00:60:1D:1E:44:78 CSWaveLAN Y 0F:FF:07 5 1441 1441 1431
00:60:1D:1E:44:4C CSWaveLAN Y 00:00:00 5 0 0 0
00:60:1D:1E:43:91 CSWaveLAN Y 0F:FF:04 5 921 921 911
00:60:1D:1E:42:14 CSWaveLAN Y 00:00:00 5 0 0 0
00:60:1D:1E:42:07 CSWaveLAN Y 00:00:00 5 0 0 0
00:60:1D:1E:43:9B CSWaveLAN Y 00:00:00 5 0 0 0
00:02:2D:3D:3A:1B IEWave Y 00:00:00 7 0 0 0
Start
Stop
Clear

Other useful tools

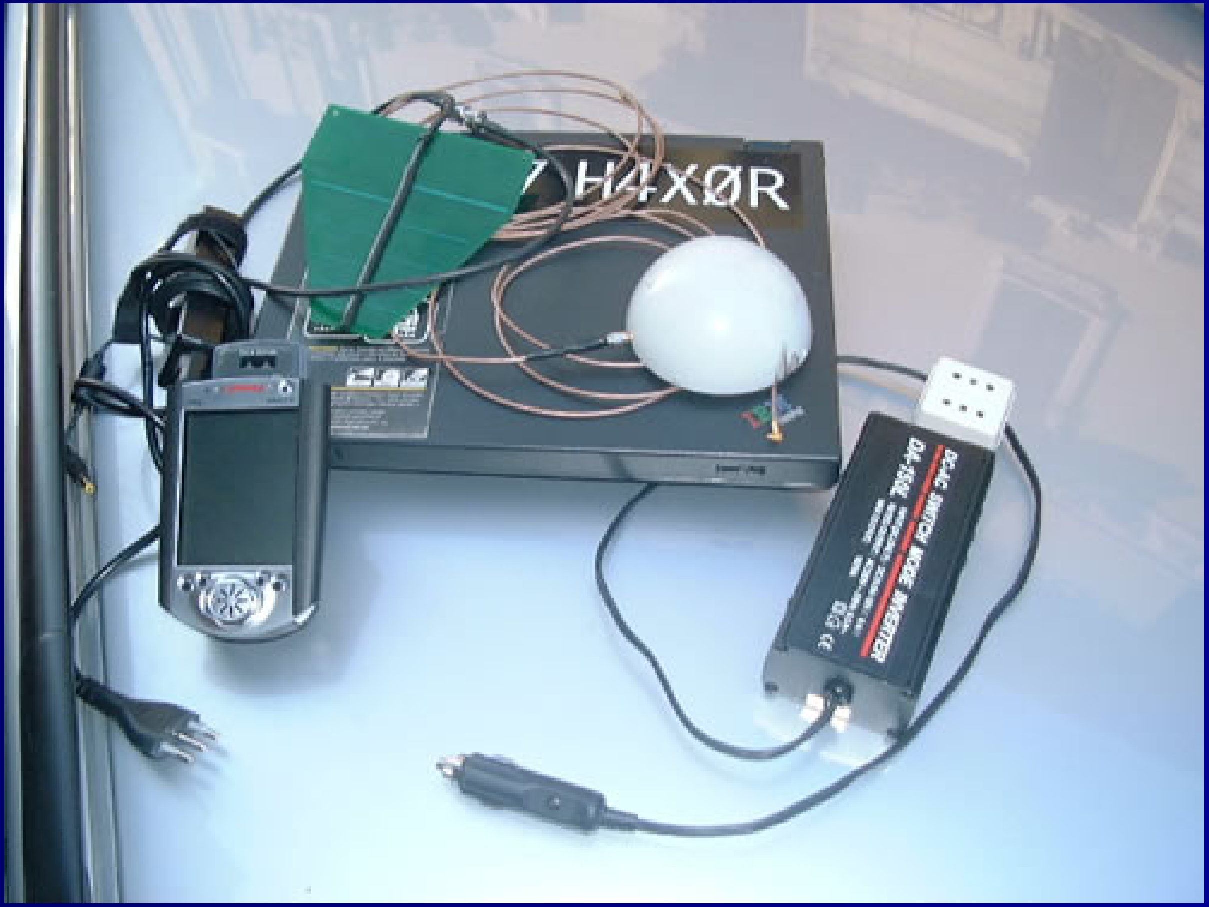
- **wavemon** detects intensity and direction of wireless signals http://www.wavemage.com/projects.html
- **Kismet** detects networks, verifies encryption type and obtains data on them such as SSID http://www.kismetwireless.net/



Level histogram
Key
[–] sig lvl (-102..10 dBm) [–] ns lvl (dBm) [ ] S-N ratio (dB)
F1info F2lhist F3aplst F4 F5 F6 F7prefs F8help F9about F10quit

```
┌-Networks---------------------------------------------------------------┐ ┌-Info--┐
|    SSID                           T W Ch  Data   LLC  Crypt  Wk Flags  | | Ntwrks |
|    linksys                        A Y 01     0    97      0   0        | |     33 |
|    HarlamNet                      A N 01     1   188      0   0        | | Pckets |
|  . Physics Network                A Y 01     9    36      3   0        | |   6145 |
|  . Travis                         A N 01     0     9      0   0        | | Cryptd |
|  . Hamilton MS2                   A N 01     4    17      0   0        | |      4 |
|  . Hamilton-Steve and Kim's rm    A N 01     0     4      0   0        | |   Weak |
|  . Wheeler MS 2                   A N 01     2     7      0   0        | |      0 |
|  . WaveLAN Network                A N 03     0    15      0   0        | |  Noise |
|  ! David's Room                   A N 01     9    82      0   0 A C    | |    138 |
|  . Hope 302                       A Y 05     3    24      0   0        | | Discrd |
|  . <no ssid>                      H N 00    17    17      0   0        | |    407 |
|  ! WirelessHomeNetwork            A N 01     0    84      0   0        | |        |
|  ! harbor+wave                    A N 06     0    27      0   0        | |        |
|  ! the new ALT                    A N 06     0    91      0   0        | | Elapsd |
|                                                                        | | 000203 |
└------------------------------------------------------------------------┘ └-H-M-S-┘
┌-Status---------------------------------------------------------------------------┐
| Removing inactive network 'Apple Network 391c2e' from display list.               |
| Detected new network 'the new ALT' bssid 00:04:5A:D0:03:F5 WEP N Ch 6             |
| Removing inactive network 'default' from display list.                            |
| Detected new network 'harbor+wave' bssid 00:40:96:44:15:C7 WEP N Ch 6             |
└-----------------------------------------------------------------------------------┘
```

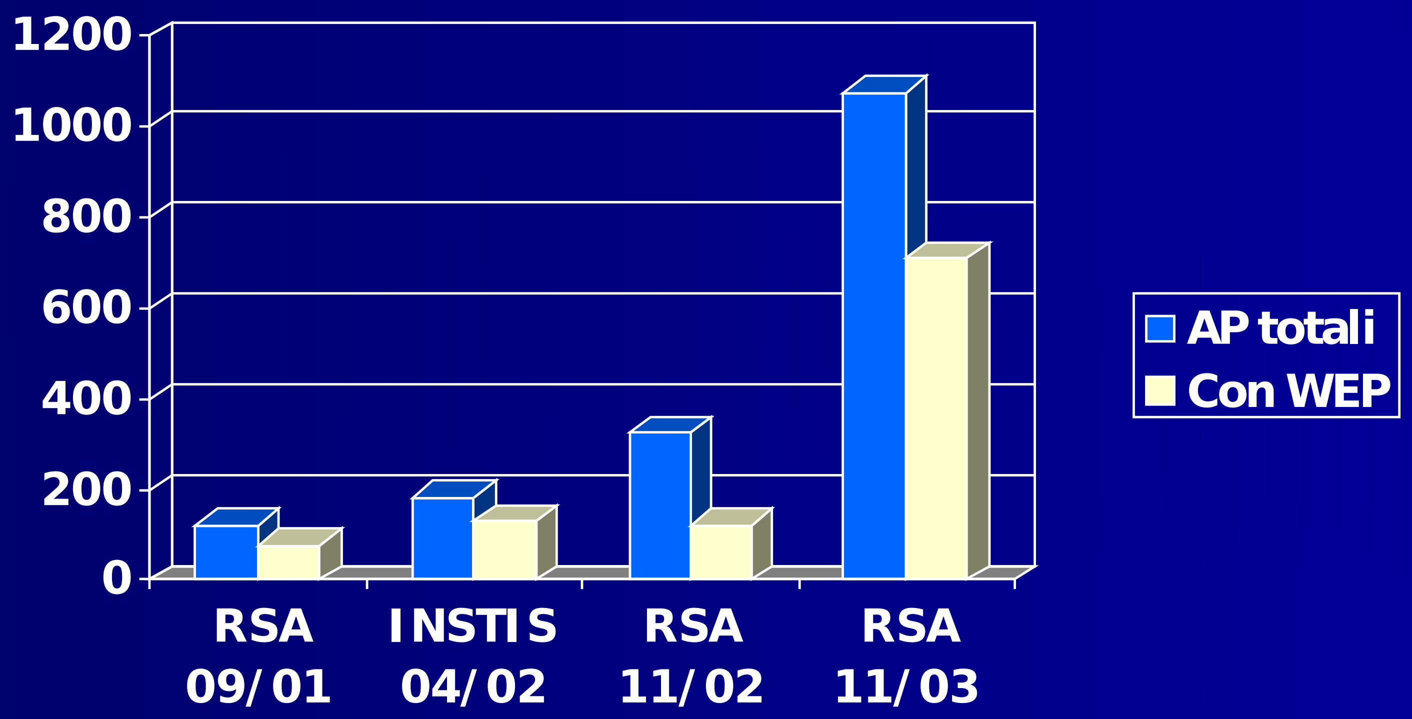
Wardriving



Wardriving: recipe

- Ingredients:
  - Car (with one driver and one wardriver, possibly...)
  - A wireless-enabled laptop or PDA
  - kismet (or NetStumbler under Windows, MacStumbler under Mac)
  - (optionally) a GPS
  - (optionally) omnidirectional antenna
  - (optionally) directional antenna
  - (optionally) inverter
  - Beware about legality of what you're doing!

Data show a trend...



IEEE 802.1x

- .1X, common to “wired” and “wireless” networks
- Proposed standard for the authentication at layer 2
- At the moment it integrates the following protocols: LEAP, EAP-TLS, EAP-TTLS and PEAP
- Uses RADIUS (Remote Access Dial-In User Service)

EAP

- Extensible Authentication Protocol (RFC 2284) an extension of PPP, adopted in 802.1x
- Allows to authenticate the user on an external server (tipically RADIUS)
- Supports a wide range of authentication mechanisms (password MD5, kerberos, One Time Password, smart card...)
- Access Point acts as mediator for the authentication
- EAP, in its original form, does NOT allow for key exchange or mutual recognition

EAP-TLS

- RFC 2716
- Uses TLS (Transport Layer Security) and digital certificate (remember, TLS = SSL v3.1)
- Automatic key generation
- Mutual authentication
- Requires existence of a PKI (Public Key Infrastructure)

EAP-TTLS

- Tunneled Transport Layer Security: client uses login+password, in a TLS tunnel
- Server uses a certificate
- Less burden for PKI

LEAP/PEAP

- LEAP is a proprietary Cisco protocol
- Uses MS-CHAPv1 for mutual authentication
- Dynamic key generation
- PEAP: Protected EAP, new standard proposed by Microsoft and Cisco, using CHAPv2
- Why ? Because CHAPv1 is weak, and as a result, LEAP is vulnerable to attack

LEAP Vulnerability

- J. Wright announced it in august 2003 at Defcon
- Uses a weakness of the challenge response mechanism of LEAP (CHAP) which was developed on the basis of NTLM passwords, which are hashed without salting
- Allows to efficiently perform a dictionary attack"
- http://asleap.sourceforge.net/ implements it
- Cisco suggested "a strong password policy" (...) and a migration to PEAP or TTLS

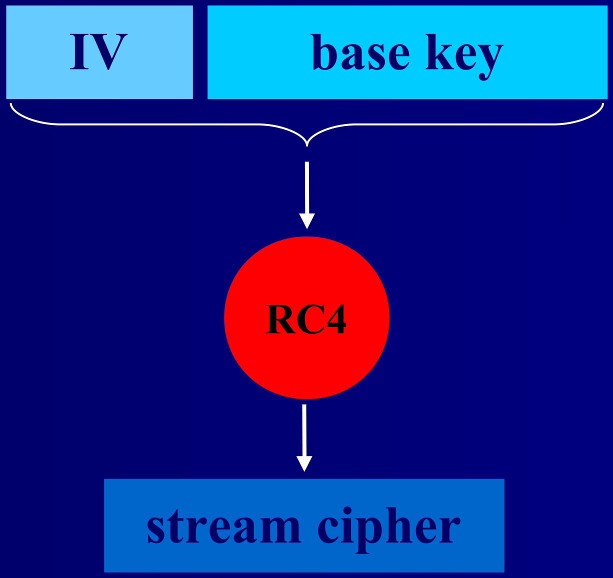
Details

- Username in cleartext
- “Two bytes vulnerability”
  - LEAP password is hashed with MD4 generating a 16 byte hash (NT_HASH)
  - 16 byte are brought to 21 adding **5 null (!!)**
  - Divided in 3*7 byte keys = 3*56 bit (just enough for DES)
  - Each subkey used to encrypt a challenge separately
  - We know the challenge (sent in clear) and we know that the third subkey has just 2 bytes (5 are fixed to NULL).
  - 2 byte = 65k combinations: can guess them
  - Can reduce the space of possible passwords (on a dictionary 3 million strong, I need to test just 45 of them)

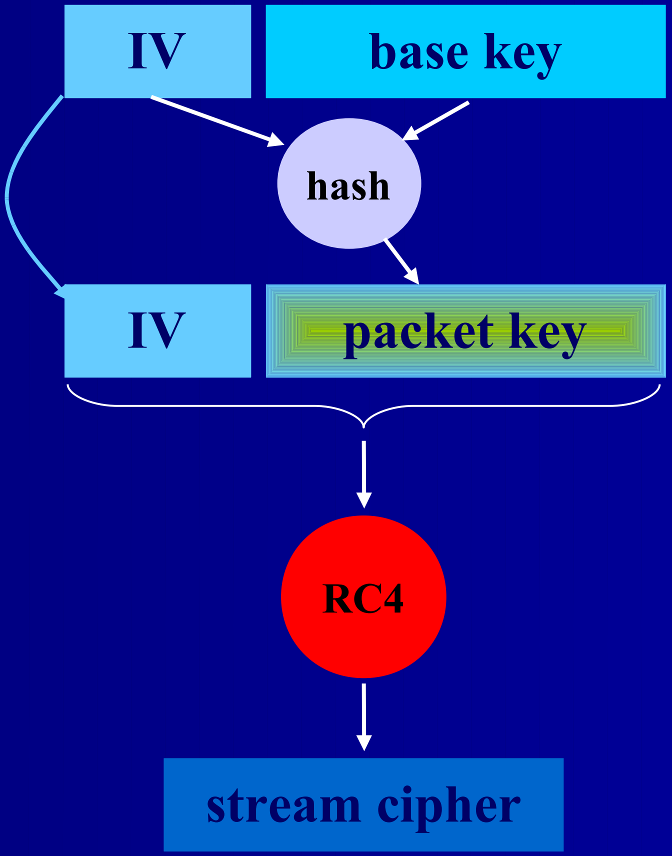
IEEE 802.11i - TKIP

- Stopgap standard to “upgrade” WEP in software (changing it to something more secure such as AES requires hw change)
- TKIP (Temporal Key Integrity Protocol) uses RC4, with a variable key

TKIP: WEP Key Hashing



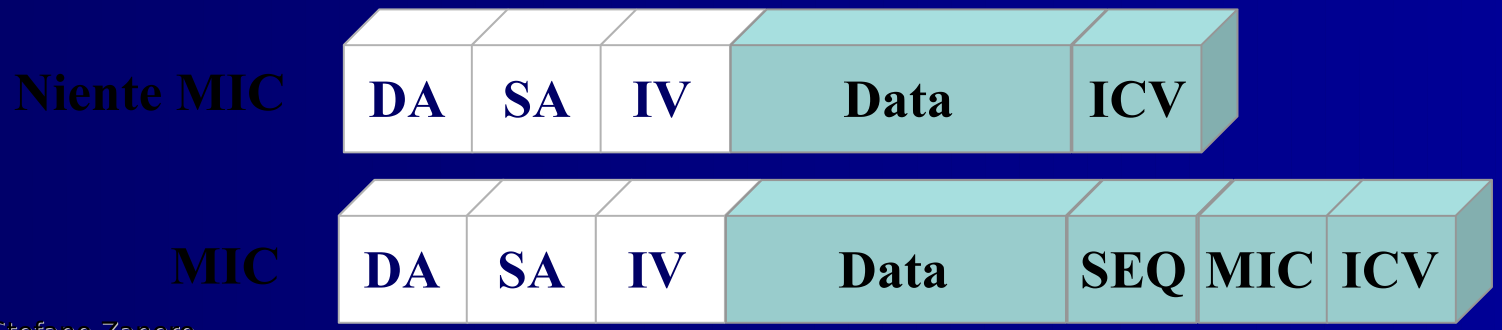
WEP: no key hashing



TKIP: key hashing

Message Integrity Check (MIC)

- MIC: Message Integrity Check, another addition of 802.11i: substituting CRC-32 with strong authentication
- MIC is a function of a random seed, of the source and dst mac addressess, and of the payload
- Since the seed and MIC are in the encrypted payload, no more blind bit flipping



Wi-Fi Protected Access (WPA)

- WPA = 802.1X-LEAP + TKIP
- WPA2 = PEAP + AES
- WPA a requirement, since August 2003, for the “Wi-Fi” logo
- WPA-PSK (shared key) is vulnerable to attack if used with TKIP (2008)

Additional “security” measures

- Using at very least WEP 128bit if WPA2 is not available
- Disable SSID broadcast and choose non-telling SSID
- MAC address filter
- Positioning the AP in the DMZ and require the use of a VPN

Bibliography

- J. Walker, “Unsafe at any key size: an analysis of WEP encapsulation”, *IEEE 802.11-00/362*, IEEE Press, 2000
- N. Borisov, I. Goldberg, D. Wagner, “Intercepting mobile communications: the insecurity of 802.11”, *Proc. 7th Ann. Intl. Conf. Mobile Computing and Networking*, ACM Press 2001
- W. A. Arbaugh, “An inductive chosen plaintext attack against WEP/WEP2”, *IEEE 802.11*, Verlag, 2001
- S. Fluhrer, I. Mantin, A. Shamir, “Weaknesses in the key scheduling algorithm of RC4”, *Proc. 8th Ann. Workshop Selected Areas in Cryptography*, Springer 2001

Bibliography

- E. Danyelyan, “802.11”: http://www.isoc.org/pubs/int/cisco-1-1.html
- Security in Wireless Networks: http://rr.sans.org/wireless/wireless_net3.php
- Using the Fluhrer, Mantin, and Shamir Attack to Break WEP: http://www.cs.rice.edu/~astubble/papers.html
- DRAFT Special Publication 800-48, Wireless Network Security: 802.11, Bluetooth, and Handheld Devices: http://csrc.nist.gov/publications/drafts/draft-sp800-48.pdf
- A Technical Comparison of TTLS and PEAP by Matthew Gast: http://www.oreillynet.com/pub/a/wireless/2002/10/17/peap.html
- Vulnerabilità di LEAP: http://home.jwu.edu/jwright/presentations/asleap-defcon.pdf