WEP

INGI2347: COMPUTER SYSTEM SECURITY (Spring 2014)

Marco Canini

Lecture slides adapted from UCL INGI2347 by Gildas Avoine, Upenn CSE331 by Steve Zdancewic, and Dan Boneh, Stanford Cryptography I@Coursera. Reproduced with permission

Announcements

- 2nd and 3rd challenge will be announced on 31 Mar
- 2nd challenge deadline: 23 Apr 23:59h
- 3rd challenge deadline: 7 May 23:59h

Plan for today

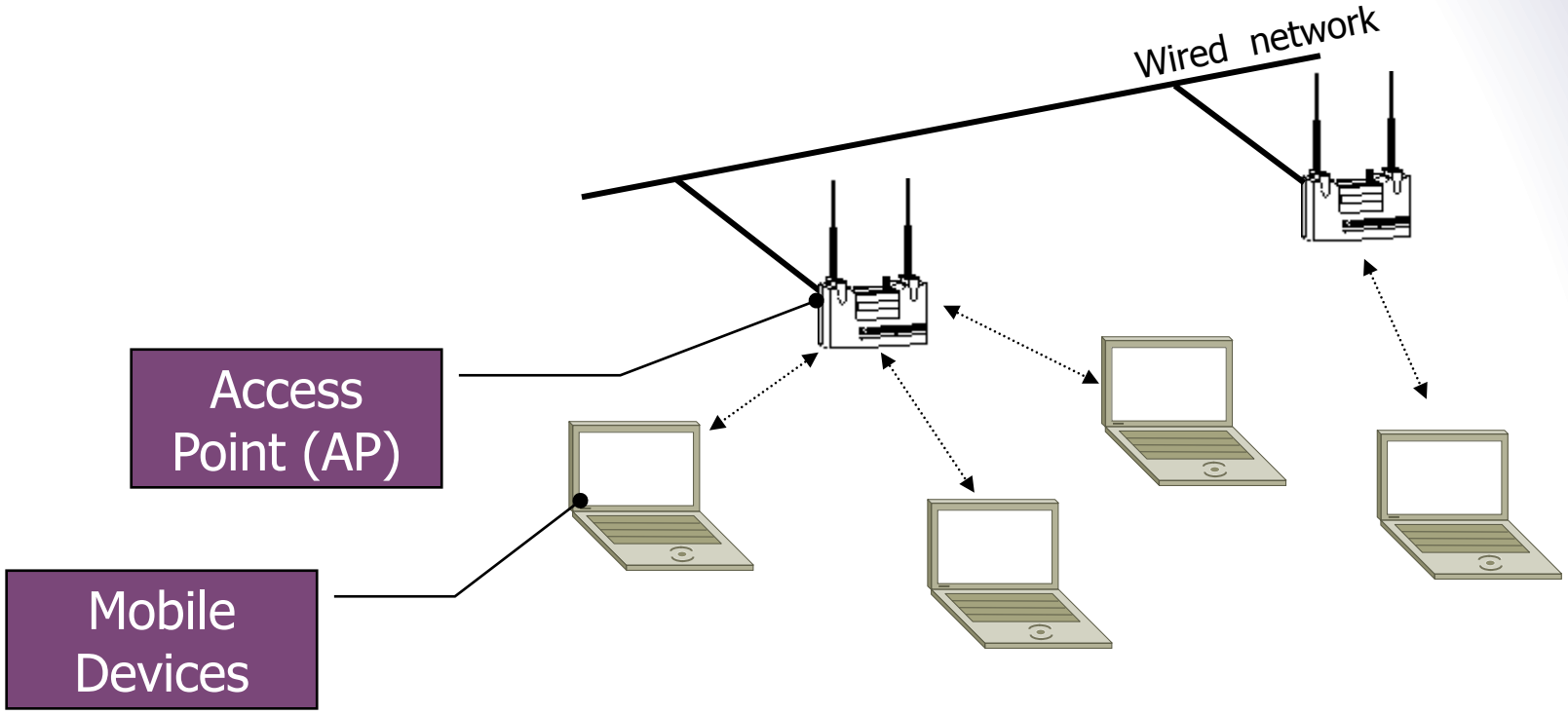
Lecture 9

- Wireless Networks intrinsic 

- Authentication
- WEP
- Attacks on WEP

WiFi: Infrastructure Mode

- Access points connect to wired network
- Multiple mobile stations per Access Point



Problem: Eavesdropping

- Wireless networking is just radio communications
  - Anyone with a radio interface can eavesdrop or inject traffic
  - Typical use inside: ~30m

Outdoor too

- Your friendly NSA can hack your WiFi over distances of up to 13 km

TOP SECRET//COMINT//REL TO USA, FVEY

NIGHTSTAND

Wireless Exploitation / Injection Tool

(TS//SI//REL) An active 802.11 wireless exploitation and injection tool for payload/exploit delivery into otherwise denied target space. NIGHTSTAND is typically used in operations where wired access to the target is not possible.

07/25/08

(TS//SI//REL) **NIGHTSTAND** - Close Access Operations • Battlefield Tested • Windows Exploitation • Standalone System

System Details

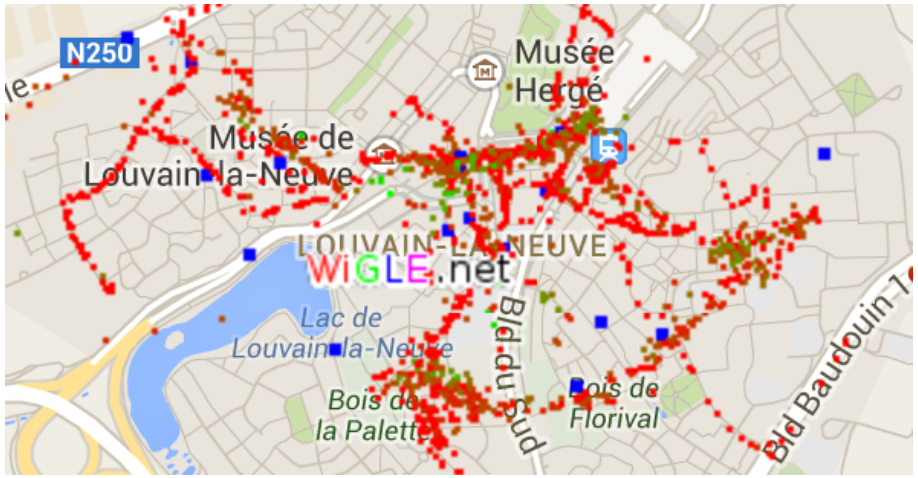
- (U//FOUO) Standalone tool currently running on an x86 laptop loaded with Linux Fedora Core 3.
- (TS//SI//REL) Exploitable Targets include Win2k, WinXP, WinXPSP1, WINXPSP2 running internet Explorer versions 5.0-6.0.
- (TS//SI//REL) NS packet injection can target one client or multiple targets on a wireless network.
- (TS//SI//REL) Attack is undetectable by the user.

NIGHTSTAND Hardware

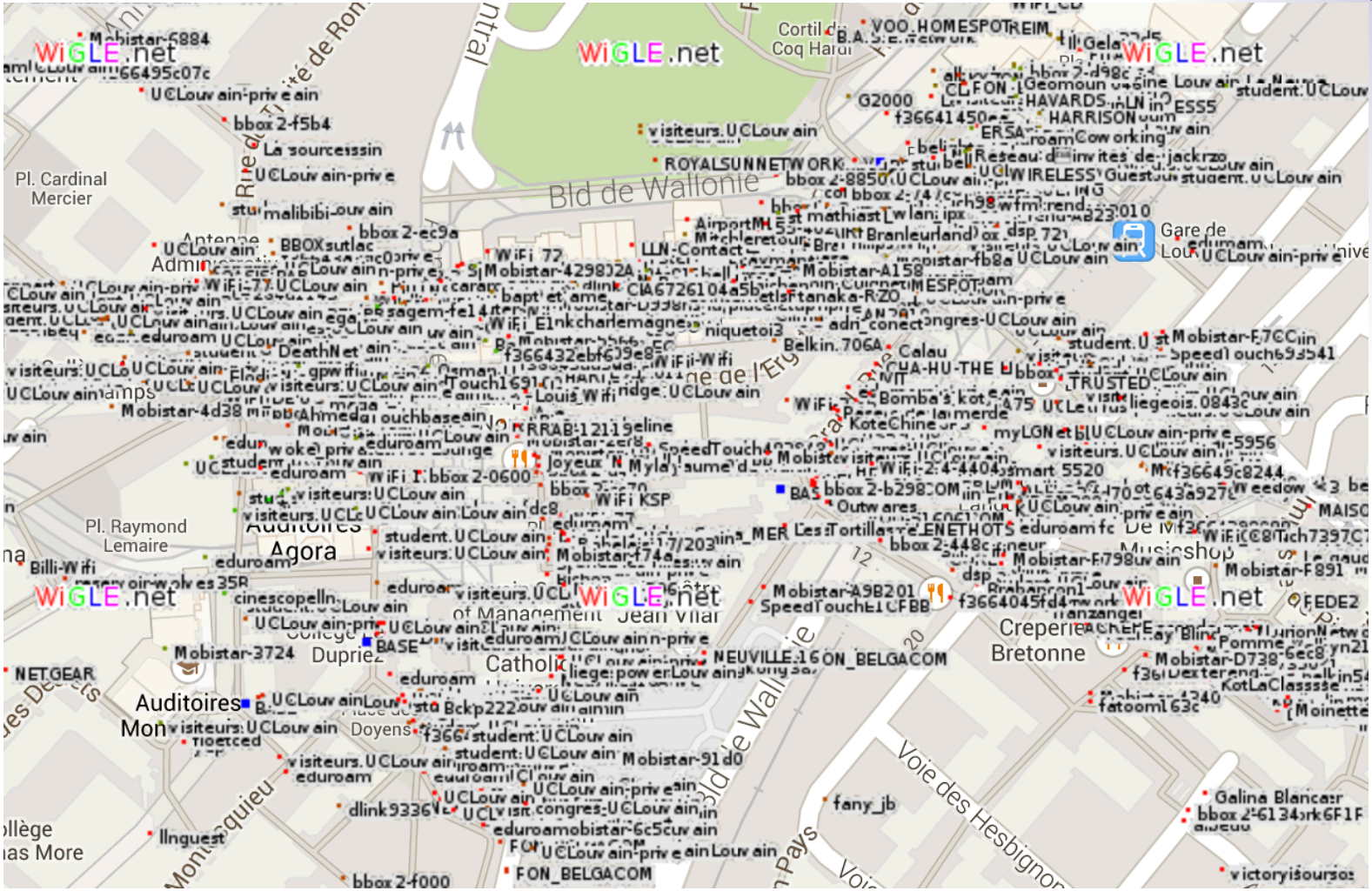
(TS//SI//REL) Use of external amplifiers and antennas in both experimental and operational scenarios have resulted in successful NIGHTSTAND attacks from as far away as eight miles under ideal environmental conditions.

Wardriving

- Discovering WiFi networks by a person in a car
  - No unauthorized access
- To wardrive:
  - Laptop
  - 802.11 adapter
  - Software
  - GPS
  - Car
- While you drive, software logs all WiFi networks around
  - E.g. www.wigle.net



Example: LLN on WiGLE

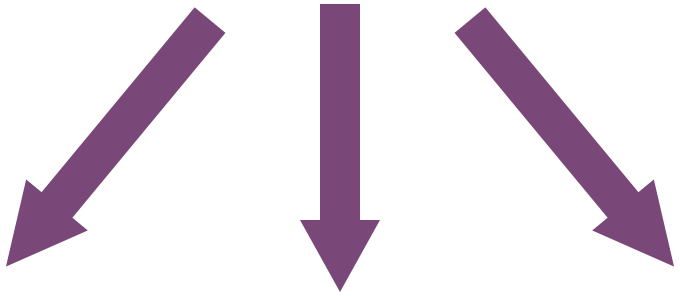


Security Needs

Easiness to intervene on a wireless network

raises

security needs



Authentication **Integrity** **Confidentiality**

+

Authentication in WiFi networks

(in general)

Authentication options

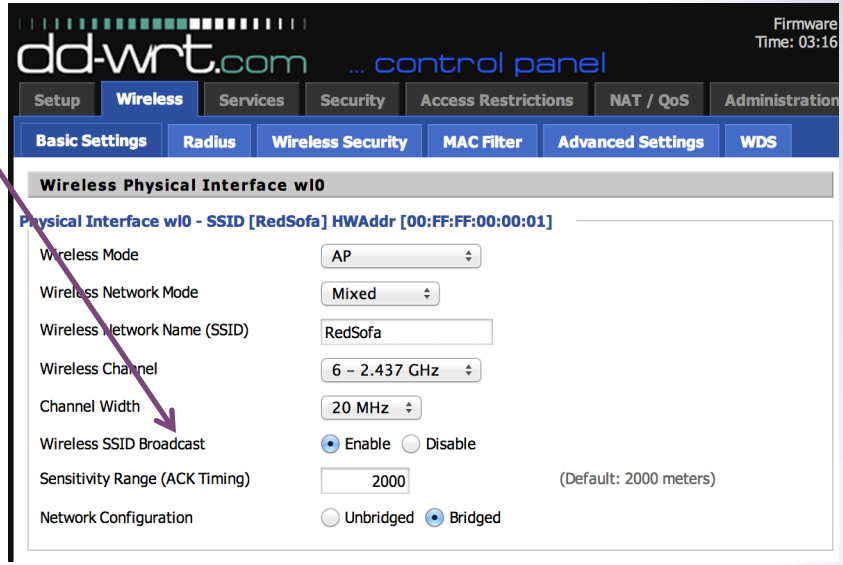
- Open systems
- Do not broadcast AP's SSID
- MAC address filter
- Cryptography (WEP, WPA, or WPA2)

Open systems

- No authentication at all
- Less and less used?
  - Usually, providers impose authentication by default
  - Public free hot spots without authentication
  - Non-free hot spots in hotels, train stations, etc.
    - High Level Authentication (e.g. RADIUS Server)
  - Communities sharing their access
    - E.g. FON (Belgacom), VOO

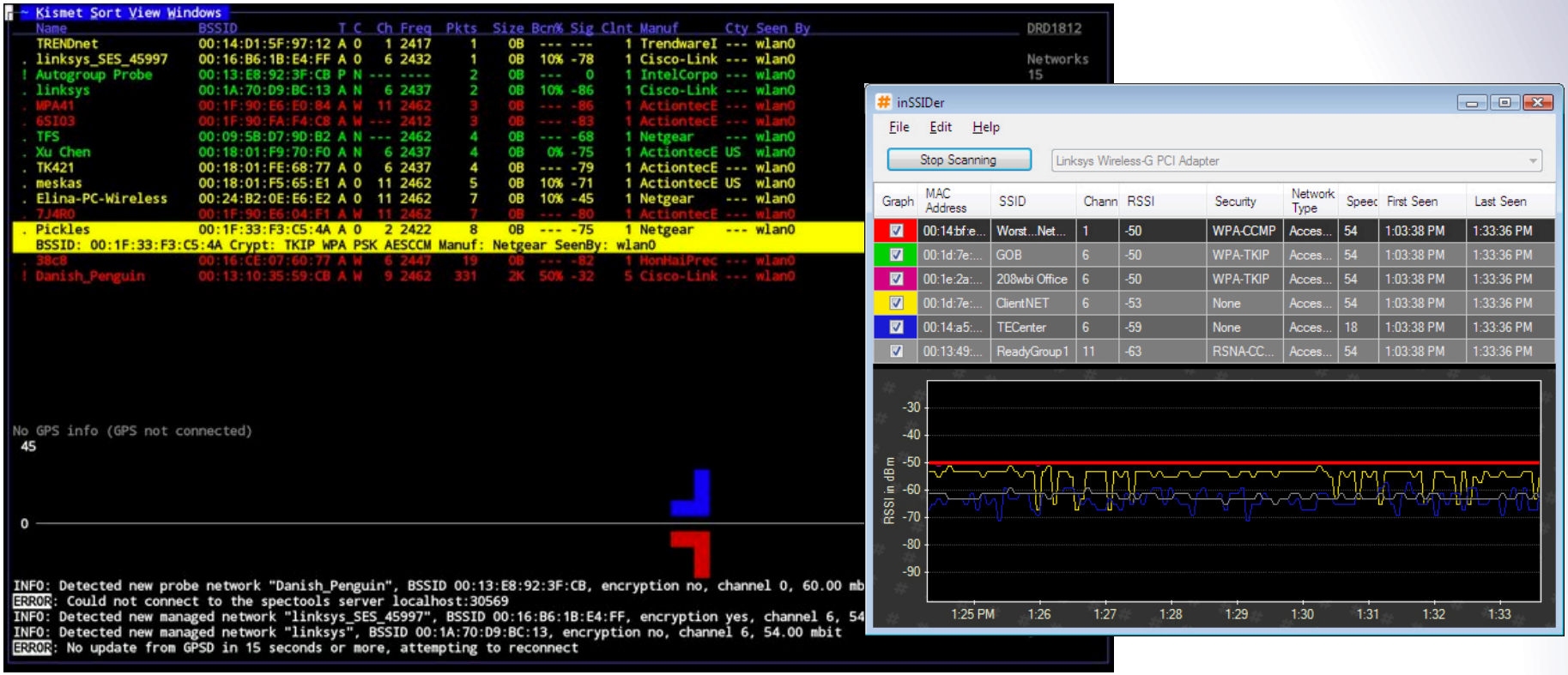
AP's SSID

- The AP broadcasts its SSID
  - Allow clients to dynamically discover the AP
- SSID be used to authenticate a client
  - By avoiding broadcasting SSID
  - Client must know the SSID
- Not secure because SSID can be eavesdropped
  - When a legitimate client connects



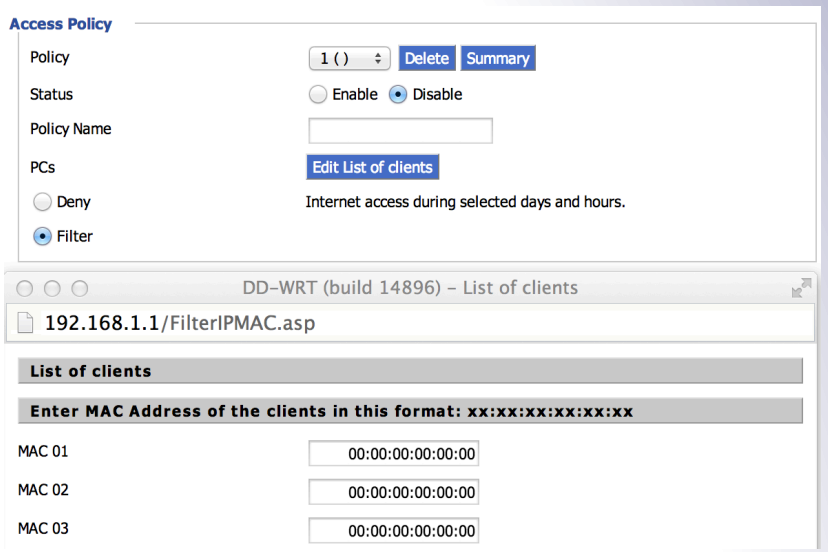
SSID in practice

- Sniff the environment with e.g. Kismet, Airodump, Network Stumbler (Windows), inSSIDer, etc.



MAC address filter

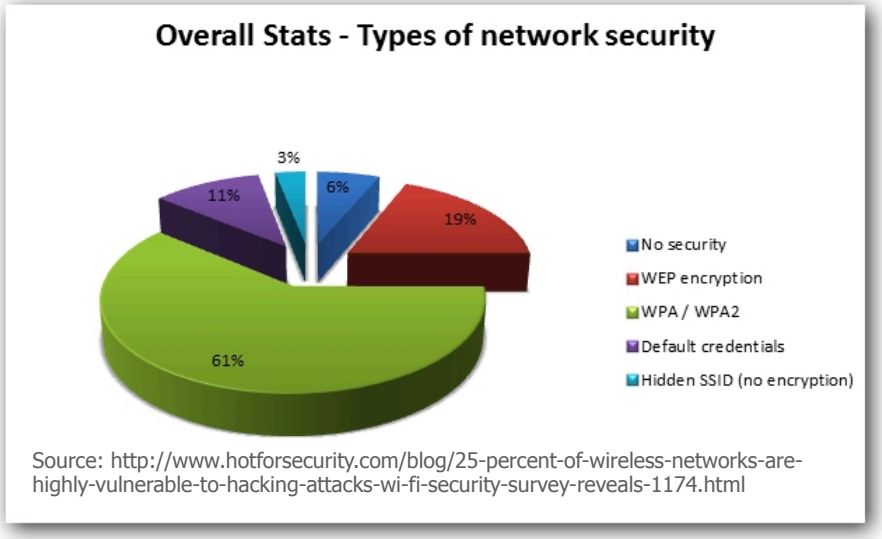
- AP stores a list of MAC addresses that are authorized to connect



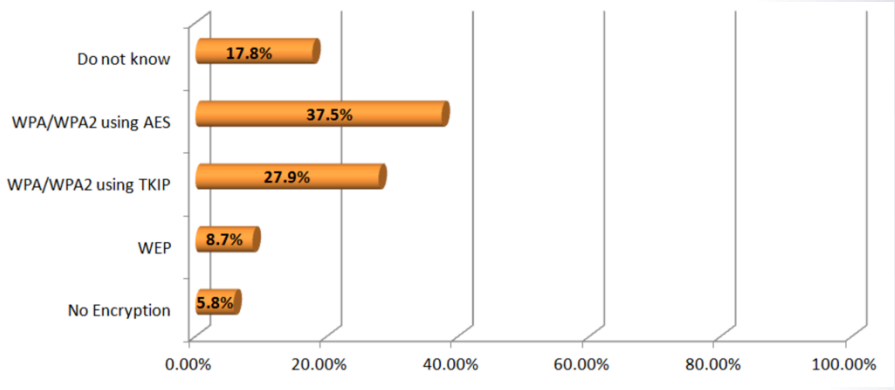
- Attacker can sniff the MAC address of a legitimate mobile station and steal it by replacing own MAC address with that MAC

Encryption

- Crypto: **WEP** or **WPA**
  - Prior to ~2010, very dire status: reports showed as low as 21-40% of the population using encryption
    Study in Boston, Chicago, NYC, SF Bay Area, Seattle, Indiana. Source: "WiFi Epidemiology: Can Your Neighbors' Router Make Yours Sick?" H. Hu, S. Myers, V. Colizza, A. Vespignani, Feb 2008.
  - Status has been improving over the years… not quite 100% yet though
  - Why?



Source: http://www.hotforsecurity.com/blog/25-percent-of-wireless-networks-are-highly-vulnerable-to-hacking-attacks-wi-fi-security-survey-reveals-1174.html



Source: http://www.safewifi.hk/files/WiFi_Adoption_and_Security_Survey_2013.pdf

Wired Equivalent Privacy (WEP)

Wired Equivalent Privacy (WEP)

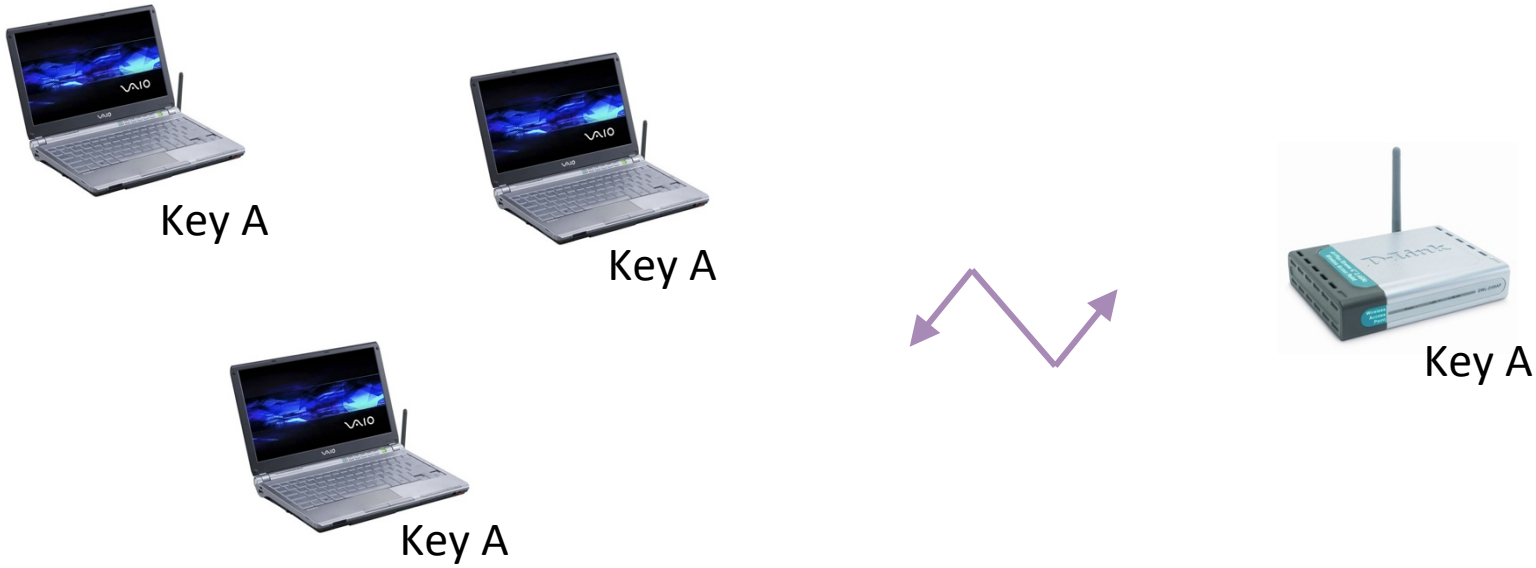
- Part of 802.11 Standard (1999)
- The stated goal of WEP is to make the wireless LAN as secure as a wired LAN

Features:

- No key management
- No protection against replay attacks
- Authentication ("shared key" user authentication)
- Confidentiality (RC4 stream cipher encryption)
- Integrity checking (CRC-32 integrity mechanism)

No Key Management

- Every mobile station and AP has the same "pre-shared" key that is used during authentication and encryption
- This key is distributed manually



No Key Management

- In practice:
  - Key is loaded in device by hand when set up
    - Often keep manufacturer's default
    - Printed under the router, in the user guide, etc.
  - Never updated again
- Same key for everybody:
  - In large networks, users should have independent secure connections
  - Just a single non-honest WLAN user can break the security
- Static key:
  - Since it is relatively easy to crack WEP encryption in a reasonably short time, keys should be changed often

Replay Attacks

Adversary can "replay" a packet she has already seen

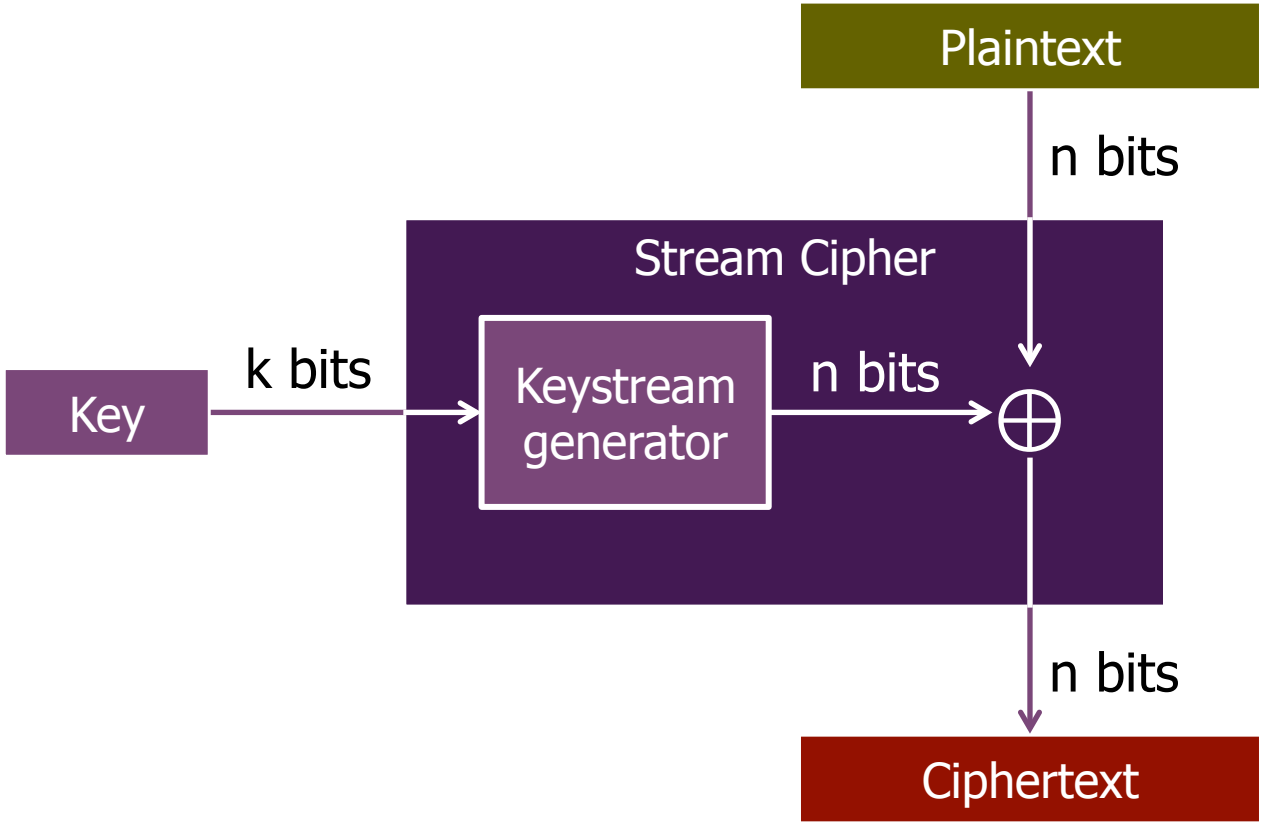
- Solutions?
  - Challenge-Response
  - Timestamp
  - Sequence Numbers

Integrity

Integrity is ensured using a CRC

- CRC does not provide cryptographic integrity
  - CRC designed to detect random errors
  - Not designed to detect artificially-constructed changes

Encryption based on Stream Cipher



The RC4 Stream Cipher

- Designed by Ron Rivest (MIT) in 1987 for RSA Labs
  - Kept as a secret trade until 1994
  - Publicly disclosed in Sep 1994 on Cypherpunks mailing list
- Bytes-oriented
  - Generate one keystream byte at each step
- Efficient in software (compared to LFSRs, Block Ciphers)
  - Encryption in software is about 10 times faster that DES
  - Simple and elegant: based on XOR
- Widely used:
  - Commercial software like MS Office, Oracle Secure SQL
  - Network protocols as SSL, IPsec, WEP
  - Copy protection: MS Xbox

RC4

Key scheduling:

- Secret key of 1 to 256 bytes to initialize a 256-byte state table S

Keystream generation:

- State table is used for subsequent generation of pseudo-random bytes

- Key was limited to 40 bits, due to export restrictions but now RC4 is generally used with a 128-bit key

Key Scheduling (KSA)

- Array K contains L bytes of key
  - L=16
  - K_i will denote $K_{(i \bmod L)}$ in what follows
- Array S always has a permutation of 0,1,…,N-1
  - $N=2^8$

```
for  i  in  0  to  N-1
   S_i = i
```

// Initialization

```
j = 0
for  i  in  0  to  N-1
   j = (j + S_i + K_i) mod N
   swap(S_i, S_j)
```

// Scrambling

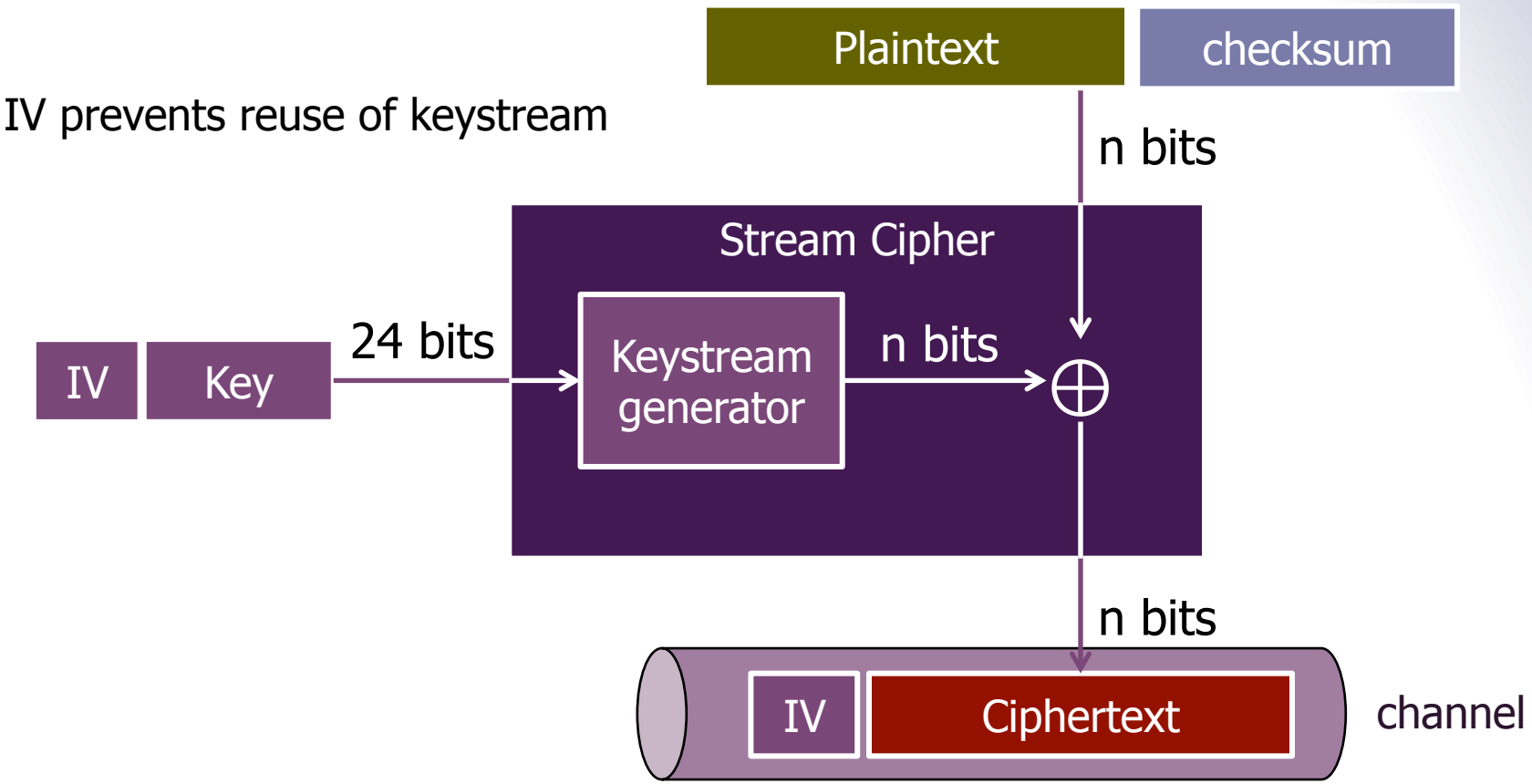
Keystream Generator (PRGA)

- Each keystream byte is generated as follows:

```
i = (i + 1) mod N
j = (j + Si) mod N
swap(Si, Sj)
output S(Si+Sj) mod N
```

// Initially i = j = 0

Encryption based on Stream Cipher



WEP Authentication

MAC address →

← Challenge (128 bytes)

Response (encrypted) →

← Status code

- Authentication is successful, if WEP decryption gives original challenge text

+

Attacks on WEP

#1 Exhaustive Search

- WEP uses 40-bit keys…
  - Due to exportation reasons
  - Easy to crack 40-bit keys in practice
  - 40-bit key + 24-bit IV = 64 bits
- Evolved version: 128 bits
  - 104-bit key + 24-bit IV = 128 bits
  - 13 characters or 26 hexs

#2 CRC Property

- CRC is a linear function w.r.t. to XOR:

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

- Attacker observes $(M \,||\, CRC(M)) \oplus K$ where K is the keystream output
  - For any ΔM, the attacker can compute $CRC(\Delta M)$
  - Hence, the attacker can compute:

$$([M \,||\, CRC(M)]) \oplus K) \oplus [\Delta M \,||\, CRC(\Delta M)]$$
$$= ([M \oplus \Delta M) \,||\, (CRC(M) \oplus CRC(\Delta M)]) \oplus K$$
$$= [(M \oplus \Delta M) \,||\, CRC(M \oplus \Delta M)] \oplus K$$

#2 Example: Δ IP Address

If the attacker knows the destination IP address:

- Attacker can change IP address in the ciphertext
- And modify CRC so that it is correct
- Then AP will decrypt and forward packet to the IP address selected by the attacker
- Requires no knowledge of the key K

#3 Keystream Reused

- WEP uses 24-bit (3 byte) IV
  - Each packet gets a new IV
  - RC4 packet key: IV pre-pended to long-term key K

If key and IV are same, then same keystream is used!

#3 Keystream Reused

Problem:

If K is never modified, IV is frequently repeated

- IV is often a counter that starts at zero
  - Hence, rebooting causes IV reuse
  - Also, there are only 16 million possible values of IV, so after intercepting enough packets, they are sure to be repeated
- There is a 50% chance of key-reuse after 2^{12} packets (birthday paradox)

#3 Keystream Reused

IV, $P \oplus RC4(K, IV)$

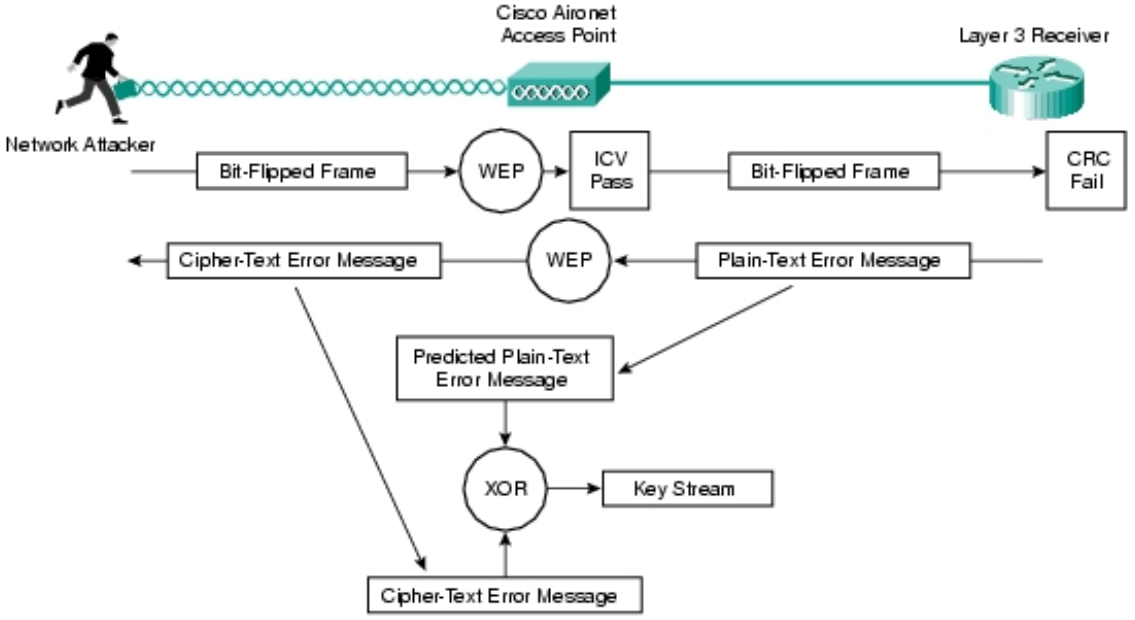
IV, $P' \oplus RC4(K, IV)$

If IV repeats, confidentiality is at risk:

- If we send two ciphertexts (C, C') using the same IV, then the XOR of plaintexts leaks ($P \oplus P' = C \oplus C'$)
- Reveal both P and P' if there is enough redundancy

#3 Getting Some Plaintext

- A bit-flipping attack is an attack on a cryptographic cipher in which the attacker can change the ciphertext in such a way as to result in a predictable change of the plaintext



Some history

- 1995 – Some security issues in RC4 (weak keys) – Roos, Wagner
- 2001 – The insecurity of 802.11 – Borisov, Goldberg, Wagner
- 2001 – Weaknesses in the key scheduling algorithm of RC4 – Fluhrer, Mantin, Shamir
- 2002 – Using the Fluhrer, Mantin, and Shamir Attack to Break WEP – Stubblefield, Ioannidis, Rubin
- 2004 – Korek, improves on the above technique and reduces the complexity of WEP cracking, proposing 17 attacks
- 2005 – Klein introduces more correlations between the RC4 key stream and the key
- 2007 – Tews, Weinmann, Pyshkin further simplify WEP Cracking

#4 Fluhrer, Mantin, and Shamir

- 2001, just 2 years after WEP was published
- Some IVs are weak, i.e., they allow to guess some internal states, leading to the key
- IV and first byte of plaintext/ciphertext must be known
  - IV is sent in the clear
  - Ciphertext is eavesdropped
  - First byte of plaintext can be easily guessed because it comes from the WEP SubNetwork Access Protocol (SNAP) header

WEP Cryptanalytic Attack

- WEP data encrypted using RC4

Packet key is IV || K – IV (3 bytes) and key K

- IV is sent in the clear (not secret)
  - New IV sent with every packet
  - Long-term key K never changed
- Assume attacker knows IVs and ciphertext, and can guess the first bytes of the plaintext
- Attacker wants to find the key K

WEP Cryptanalytic Attack

- We denote the RC4 key **bytes** as:
  - K_0, K_1, K_2, K_3, K_4, K_5,…
  - Where IV = (K_0, K_1, K_2), which the attacker knows
  - Attacker wants to find K_3, K_4, K_5,…

RC4 Steps

- KSA (Key-Scheduling Algorithm)
  - Initialization
  - Scrambling
- PRGA (Pseudo-Random Generation Algorithm)

WEP Cryptanalytic Attack

```
for  i  in  0  to  N-1
   Si = i
```

// Initialization

i	0	1	2	3	4	5	6	…
S_i	0	1	2	3	4	5	6	…

WEP Cryptanalytic Attack

```
j = 0
for  i  in  0  to  N-1
   j = (j + Si + Ki) mod N
   swap(Si, Sj)
```

// Scrambling

i	0	1	2	3	…
K_i	4	8	242	254	…

Example of key

initial S

i=0, j=0+S_0+K_0=4

i=1, j=4+S_1+K_1=4+1+8=13

i	0	1	2	3	4	…	…	N-1
S_i	0	1	2	3	4	…	…	N-1
S_i	4	1	2	3	0	…	…	N-1
S_i	4	13	2	3	0	…	…	N-1

Example from "Break WEP Faster with Statistical Analysis", by Rafik Chaabouni, 2006

WEP Cryptanalytic Attack

```
i = (i + 1) mod N
j = (j + Si) mod N
swap(Si, Sj)
output S(Si+Sj) mod N
```

// Initially i = j = 0

i	0	1	2	3	…
K_i	4	8	242	254	…

Example of key

After KSA (Assumption)

i=1, j=0+2=2

i	0	1	2	3	4	5	…	N-1
S_i	4	2	3	13	43	7	…	N-1
S_i	4	3	2	13	43	7	…	N-1

First outputted byte is $S_{(S_1+S_2)}=S_{(3+2)}=S_5=7$

WEP Cryptanalytic Attack

- Attack due to Fluhrer, Mantin, and Shamir

Attacker watches IVs until seeing IV of the form:

IV = (K_0, K_1, K_2) = (3, 255, X)

- Where X can be anything (attacker knows X)

- Then RC4 key for this packet is

  key = (3, 255, X, K_3, K_4, K_5, …)

- Attacker wants to find (K_3, K_4, K_5, …)

WEP Cryptanalytic Attack

```
j = 0
for  i  in  0  to  N-1
   j = (j + Si + Ki) mod N
   swap(Si, Sj)
```

// Scrambling

i	0	1	2	3	4	…
K_i	3	255	X	K_3	K_4	…

	i\S	0	1	2	3	4	…	5+X	…	6+X+K_3	…
initial state	init	0	1	2	3	4	…	5+X	…	6+X+K_3	…
i=0, j=0+S_0+K_0=0+0+3=3	i=0	3	1	2	0	4	…	5+X	…	6+X+K_3	…
i=1, j=3+S_1+K_1=3+1+255=3[N]	i=1	3	0	2	1	4	…	5+X	…	6+X+K_3	…
i=2, j=3+S_2+K_2=3+2+X=5+X	i=2	3	0	5+X	1	4	…	2	…	6+X+K_3	…
i=3, j=(5+X)+(1)+K_3=6+X+K_3	i=3	3	0	5+X	6+X+K_3	4	…	2	…	1	…

WEP Cryptanalytic Attack

- So far we have only considered the first 4 steps of initialization, i = 0,1,2,3
  - In reality, there are 256 steps
- For now, assume that initialization stops after i = 3
- So, the keystream outputted by PRGA is:

```
i = (i + 1) mod N
j = (j + Si) mod N
swap(Si, Sj)
output S(Si+Sj) mod N
```

```
= 1                    // Initially i = j = 0
= S1 = 0
swap(S1, S0)
output S3 = 6+X+K3
```

WEP Cryptanalytic Attack

- Keystream byte $Y = 6+X+K_3$
- If Y is known, we can solve for K_3 since:

$K_3 = (Y-6-X) \bmod N$

- But initialization does not stop at i=3 …
- So can this "attack" really work?
- If elements at 0,1 and 3 are not swapped in remaining initialization steps, attack works!

WEP Cryptanalytic Attack

- For remaining initialization steps…
  - We have i=4,5,6,… so index i will not affect anything at indices 0,1 or 3
  - But what about index j?
- Pretend index j selected at random
  - At each step, probability is 253/256 that $j \notin \{0,1,3\}$
  - There are 252 steps after i=3
- Probability that 0,1 and 3 are not affected by index j after step i=3 is

$(253/256)^{252} = 0.0513$

WEP Cryptanalytic Attack

Can the attacker really recover the key?

- After seeing enough IVs can obtain K_3

Suppose attacker got K_3

- How to find K_4?

Consider IVs of the form: IV = (4, 255, X).

- Then after initialization step i=4, one can show that:

Keystream byte $Y = S_4 = 10+X+K_3+K_4$

WEP Cryptanalytic Attack

- If enough IVs are available
  - And corresponding 1st keystream bytes are known
- Then attacker can recover the key
  - Find K_3 then K_4 then K_5 and so on…
- Get entire key, regardless of length

WEP Cryptanalytic Attack

- 4 million IVs to recover a 128-bit key
- Number of IVs is linear with the key-length
  - (vs exponential)
- Key is revealed sequentially byte after byte

Further Attacks

- Korek – 2004
  - Proposed 17 attacks based on FMS
  - New classes of weak IVs
  - 1 million IVs

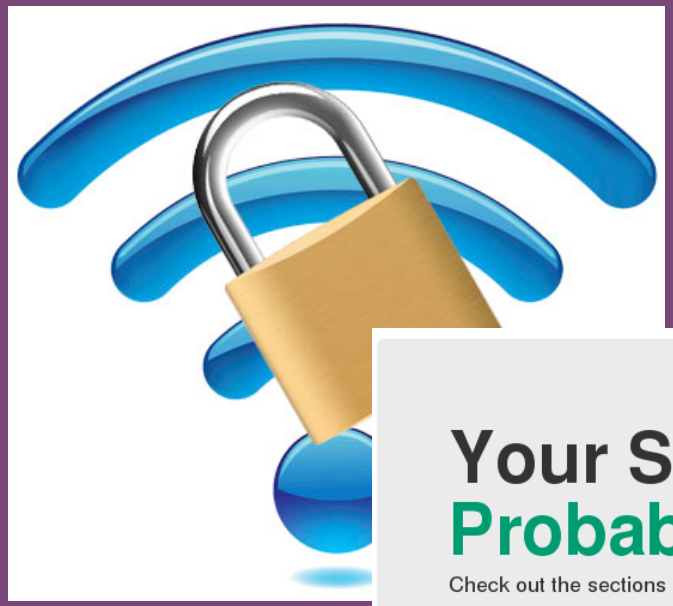
- Tews, Weinmann, Pyshkin (PTW) – 2007
  - Still new classes
  - 80,000 IVs
  - Variant by Vaudenay/Vuagnoux 2007 (32,000 IVs)
  - Key bytes are no longer necessarily guessed sequentially

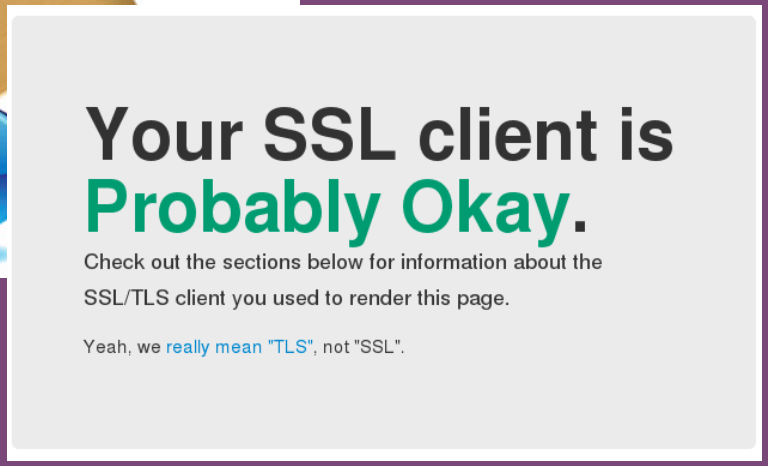
Downloadable Tools

- AirCrack-ng
  - http://www.aircrack-ng.org
  - Implement Korek, PTW (needs ARP flooding)
- WepCrack
  - http://sourceforge.net/projects/wepcrack/
  - "WEPCrack is a tool that cracks 802.11 WEP encryption keys using the latest discovered weakness of RC4 key scheduling"
  - Last version: Oct 2004
- AirSnort
  - http://airsnort.shmoo.com/
  - Last update: 2005 (no longer supported)
  - Implement Korek's attacks

Any questions?

Stay tuned





+

Next time you will learn about

WPA | SSL/TLS