

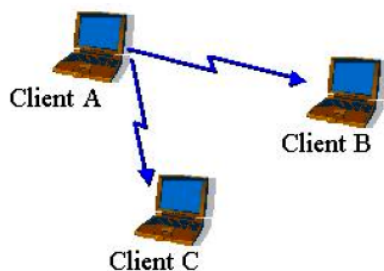
Wireless security (WEP)

9/01/10

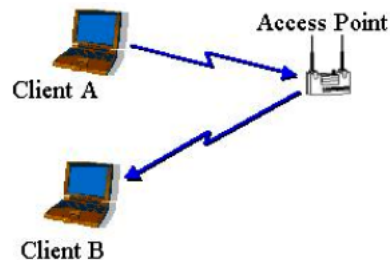
EJ Jung

802.11b Overview

- Standard for wireless networks
 - Approved by IEEE in 1999
- Two modes: **infrastructure** and **ad hoc**



IBSS (ad hoc) mode
Independent Basic Service Set



BSS (infrastructure) mode
Basic Service Set

Access Point SSID

- Service Set Identifier (SSID) differentiates one access point from another
 - By default, access point broadcasts its SSID in plaintext “beacon frames” every few seconds
- Default SSIDs are easily guessable
 - Linksys defaults to “linksys”, Cisco to “tsunami”, etc.
 - This gives away the fact that access point is active
- Access point settings can be changed to prevent it from announcing its presence in beacon frames and from using an easily guessable SSID
 - But then every user must know SSID in advance

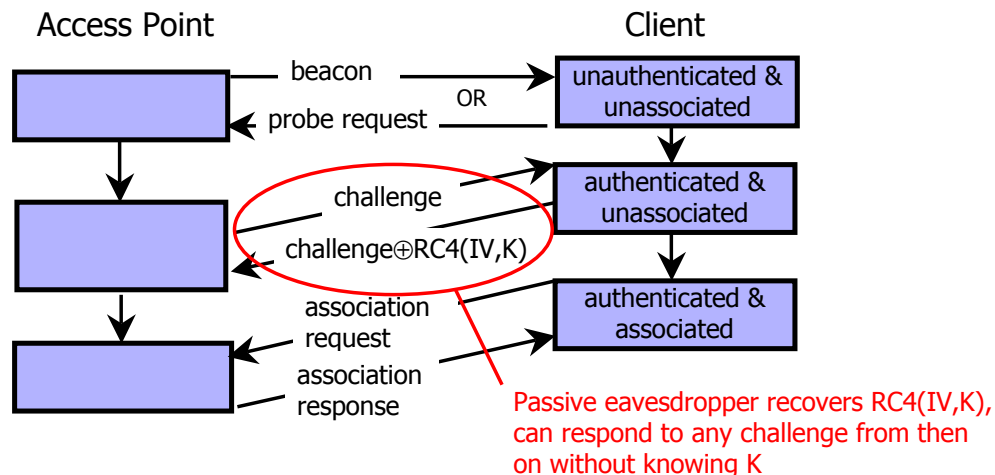
Wired Equivalent Protocol (WEP)

- Special-purpose protocol for 802.11b
 - Intended to make wireless as secure as wired network
- Goals: confidentiality, integrity, authentication
- Assumes that a secret key is shared between the access point and clients
- Uses RC4 stream cipher seeded with 24-bit initialization vector and 40-bit key
 - Terrible design choice for wireless environment
 - In SSL, we will see how RC4 can be used properly

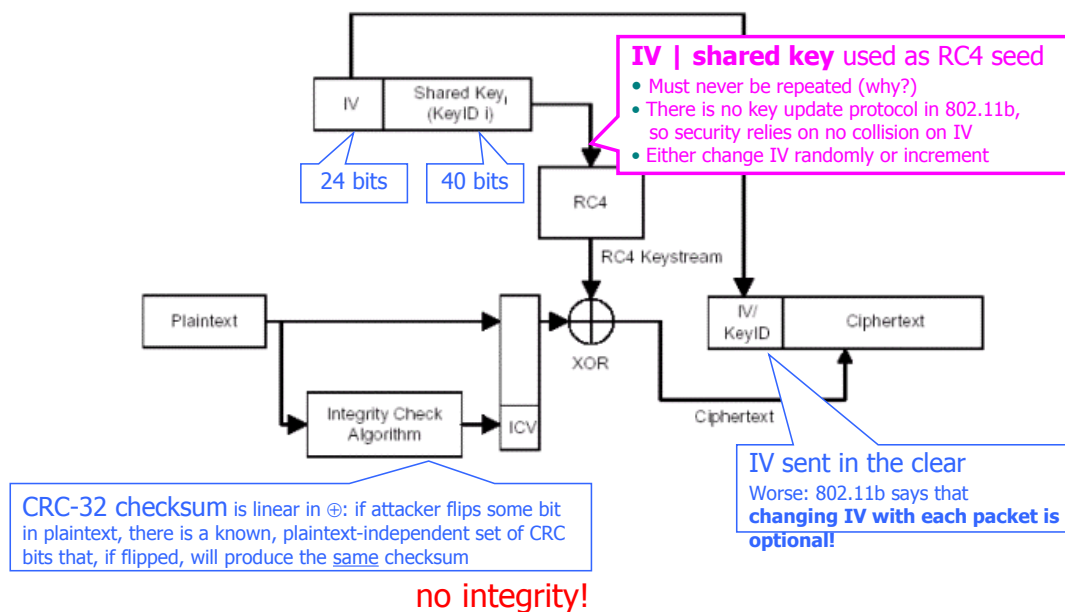
- **None** of security goals are met
- “Insecurity of 802.11” [BGW’01]
 - Keystream reuse [confidentiality]
 - CRC attacks [integrity]
 - Authentication spoofing [access control]
 - IP redirection & TCP reaction attacks [confidentiality]
- “Inductive chosen plaintext attack” [Arb’01]
 - CRC attack [confidentiality]
- “Weaknesses in RC4 key scheduling” [FMS’01]
 - RC4 weakness [confidentiality]

Shared-Key Authentication

Prior to communicating data, access point may require client to authenticate



How WEP Works



Why RC4 is a Bad Choice for WEP

- Stream ciphers require synchronization of key streams on both ends of connection
 - This is not suitable when packet losses are common
- WEP solution: a separate seed(IV) for each packet
 - Can decrypt a packet even if a previous packet was lost
- But number of possible seeds is not large enough!
 - RC4 seed = 24-bit initialization vector + fixed key
 - Assuming 1500-byte packets at 11 Mbps, 2^{24} possible IVs will be exhausted in about 5 hours
- Seed reuse is **deadly** for stream ciphers

Recovering Keystream

- Get access point to encrypt a known plaintext
 - Send spam, access point will encrypt and forward it
 - Get victim to send an email with known content
- If attacker knows plaintext, it is easy to recover keystream from ciphertext
 - $C \oplus M = (M \oplus \text{RC4}(\text{IV}, \text{key})) \oplus M = \text{RC4}(\text{IV}, \text{key})$
 - Not a problem if this keystream is not re-used
- Even if attacker doesn't know plaintext, he can exploit regularities (plaintexts are not random)
 - For example, IP packet structure is very regular

Keystream Will Be Re-Used

- In WEP, repeated IV means repeated keystream
- Busy network will repeat IVs often
 - Many cards reset IV to 0 when re-booted, then increment by 1 \Rightarrow expect re-use of low-value IVs
 - If IVs are chosen randomly, expect repetition in $O(2^{12})$ due to birthday paradox (similar to hash collisions)
- Recover keystream for each IV, store in a table
 - $(\text{KnownM} \oplus \text{RC4}(\text{IV}, \text{key})) \oplus \text{KnownM} = \text{RC4}(\text{IV}, \text{key})$
 - Even if don't know M, can exploit regularities
- Wait for IV to repeat, decrypt and enjoy plaintext
 - $(M' \oplus \text{RC4}(\text{IV}, \text{key})) \oplus \text{RC4}(\text{IV}, \text{key}) = M'$

It Gets Worse

- Misuse of RC4 in WEP is a design flaw with no fix
 - Longer keys do not help!
 - The problem is re-use of IVs, their size is fixed (24 bits)
 - Attacks are passive and very difficult to detect
- Perfect target for Fluhrer et al. attack on RC4
 - Attack requires known IVs of a special form
 - WEP sends IVs in plaintext
 - Generating IVs as counters or random numbers will produce enough “special” IVs in a matter of hours
- This results in key recovery (not just keystream)
 - Can decrypt even ciphertexts whose IV is unique

Fragmentation attack [Bittau, Handley, Lackey '06]

- WEP runs on top of 802.11x
- Fragmentation improves performance in noisy environment
 - 802.11b, 802.11g run on the same bandwidth with cordless phones, microwaves

Keystream retrieval

- Content of first 8 bytes is known in 802.11

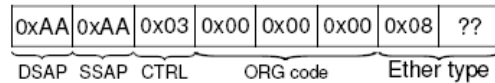


Figure 2. LLC/SNAP header contained in practically all 802.11 data frames.

- The attacker can retrieve 8 bytes of keystream
 - $\text{cleartext} \oplus (\text{cleartext} \oplus \text{RC4}(\text{IV}, \text{key})) = \text{RC4}(\text{IV}, \text{key})$

Fragment and then encrypt

- 802.11 allows 16 fragmentation
 - the same keystream may be used for 16 times
- The attacker can inject any 64 bytes
 - $64 = 16 \text{ fragments} * (8 - 4 \text{ bytes CRC})$

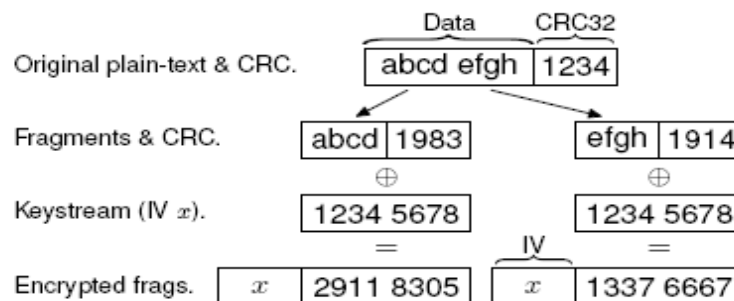


Figure 3. Transmitting a single logical packet in multiple 802.11 fragments.

Decrypt real-time

- Ask AP to forward to a known host
 - WEP only encrypts the packets between clients and AP
 - Attacker pre-pends the packet with the IP address of the known host
 - AP decrypts the packets and forwards the message in clear

- This attack works no matter what the key size is, no matter how often the key changes

Weak Countermeasures

- Run VPN on top of wireless
 - Treat wireless as you would an insecure wired network
 - VPNs have their own security and performance issues
 - Compromise of one client may compromise entire network
- Hide SSID of your access point
 - Still, raw packets will reveal SSID (it is not encrypted!)
- Have each access point maintain a list of network cards addresses that are allowed to connect to it
 - Infeasible for large networks
 - Attacker can sniff a packet from a legitimate card, then re-code (spoof) his card to use a legitimate address

- <http://tapir.cs.ucl.ac.uk/bittau-wep.pdf>
- <http://www.aircrack-ng.org/doku.php>
- <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>
- <http://www.securityfocus.com/infocus/1814>