Jonathan von Schroeder

THE INSECURITY OF 802.11

Based on Intercepting Mobile Communications: The Insecurity of 802.11 by N. Borisov, I. Goldberg, and D. Wagner,
Attacks on the WEP protocol by Erik Tews

THE IEEE 802.11 STANDARD

- Describes a protocol for
  - Communication with max. speeds between 2 Mbit/s (802.11) and 300 Mbit/s (802.11n) at a frequency band at about 2.4 GHz
  - Communication with a max. speed of 54 Mbit/s in 5 GHz frequency band
- And a simple security protocol called *Wired Equivalent Privacy* (WEP)

WHAT IS WEP?

- Intended to protect link-layer communication from eavesdropping
  - Only protects data frames
- Allows up to four different keys (more in some implementations) which are used for encryption
  - Usually only one key is used
  - Will be called Root Key (RK) in the following
- Primary security goals are
  1. **Confidentiality**
  2. **Access control**
     - 802.11 includes an optional feature to discard all improperly encrypted traffic
  3. **Data integrity**

WHAT IS WEP?

- Depends on the impracticability of a brute-force attack on the key
  - The standard specifies the use of 40bit keys (due to US-Government export restrictions at the time of drafting)
    - Most vendors offer „128 Bit-Encryption“ (104 Bit key + 24 Bit Initialization Vector (IV))

HOW DOES WEP WORK?

- WEP relies on a stream cipher called RC4, that generates an arbitrary length key stream from a RK and an IV, for encryption
  - Although there are many known attacks on RC4 itself (including key-recovery attacks) we‘ll mainly focus on flaws in WEP not related to RC4

HOW DOES WEP WORK?

- Sending the payload M using WEP includes following steps
  - The sender picks an IV
    - By using a pseudo-random number generator
    - By remembering the last IV and (interpreting it as a number) adding 1 to it
      - On initiliazation or when the highest possible number has been reached the IV is either reset to zero or a random number
  - crc(M), which is called the Integrity Check Value (ICV), is calculated
  - The IV and the RK are fed into RC4 to produce a key stream X of the combined length of M and the ICV
  - The cypher text $<M,crc(M)> \oplus X$ is calculated
  - The cypher text, the IV and few additional headers are sent over the radio-link

HOW DOES WEP WORK?

- To recover the payload M from the received Packet X=<Headers,IV,C> the receiver
  - Decrypts E by recovering the Key-Stream from IV and his knowledge of RK
    - This is possible because RC4 generates the same keystream for the same (IV,RK) tuple
    - $<M,ICV>=C \oplus RC4(IV,RK)$
  - „Verifies“ the integrity of the payload M by checking $crc(M)=ICV$

DATA INTEGRITY

WHY IS WEP INSECURE? – DATA INTEGRITY

- The CRC checksum is used to „validate“ a message
  - It is **insufficient for message authentication** because it does not hold ind-cca1 and is only meant to detect random errors
  - A general property of CRC checksums is, that it is a linear function of the message i.e. $crc(x \oplus y) = crc(x) \oplus crc(y)$ holds for all x,y

WHY CRC FAILS TO PROTECT THE INTEGRITY OF THE MESSAGE

- Message Modification
  - Claim: An intercepted Packet $X = <Headers, IV, <M, crc(M)> \oplus RC4(IV, RK)>$ can be modified such that the payload of X‘ decrypts to M‘ instead of M without disrupting the checksum
    - Choose $<\Delta, crc(\Delta)>$ such that $M' = \Delta \oplus M$
    - Then $X' = <Headers, IV, <M, crc(M)> \oplus RC4(IV, RK) \oplus <\Delta, crc(\Delta)>>$

WHY CRC FAILS TO PROTECT THE INTEGRITY OF THE MESSAGE

$= \langle Headers, IV, \langle M \oplus \Delta, crc(M) \oplus crc(\Delta) \rangle \oplus RC4(IV,RK) \rangle$

$= \langle Headers, IV, \langle M', crc(M \oplus \Delta) \rangle \oplus RC4(IV,RK) \rangle$

$= \langle Headers, IV, \langle M', crc(M') \rangle \oplus RC4(IV,RK) \rangle$

- Message Injection
  - Since CRC checksums are unkeyed as soon a key-stream with corresponding IV is known the adversary can inject arbitrary packets
  - The IV can be reused for all packets sent

ACCESS CONTROL

WHY IS WEP INSECURE? – ACCESS CONTROL

- WEP includes a challenge-response authentatication method
  - The AP sends the client an unencrypted 128-Bit string (the challenge)
  - The client responds by encrypting the challenge
  - If the challenge was encrypted correctly by the client the AP considers the authentication successful
  - This is of course insecure because the content of the response can be easily changed since the plain-text is known and thus a client that has spoofed an authentication can authenticate itself

CONFIDENTIALITY

WHY IS WEP INSECURE? – CONFIDENTIALITY

- Keystream reuse (i.e. encrypting two texts using the same IV and RK) can reveal information about both plaintexts
  - Let $C_1 = M_1 \oplus RC4(IV,RK)$ and $C_2 = M_2 \oplus RC4(IV,RK)$
    Then $C_1 \oplus C_2 =$
    $(M_1 \oplus RC4(IV,RK)) \oplus (M_2 \oplus RC4(IV,RK))$
    $= M_1 \oplus M_2 \oplus RC4(IV,RK) \oplus RC4(IV,RK) = M_1 \oplus M_2$
  - If M_1 is known M_2 can be directly computed
  - If neither M_1 nor M_2 are known there are still many techniques including frequency-analysis to recover the plaintexts
    - The Problem becomes easier if more than two plain-texts encrypted with the same Rk and IV are known

DOES KEYSTREAM REUSE ACTUALLY OCCUR?

- WEP uses a per-packet IV
  - The standard does not require a different IV for every packet
    - A compliant implementation can reuse the same IV for all packets
  - Many PCMCIA cards reset their stored IV to 0 on every initialization
  - The IV is only 24bit long and thus a busy access point sending 1500 byte packets achieving 5Mbps avg. throughput will exhaust the available IV-space in less than half a day
  - If the IV is selected randomly an IV-collision is expected to occur after only 5000 packets (which is due to the birthday-paradox)

WHAT CAN BE DONE WHEN IV-COLLISIONS ARE FOUND?

- Either parts of the plaintext are known (well-defined protocols like IP,TCP) or
- the attacker may cause well known plaintext to be transmitted by for example sending IP-traffic from the WEB
- Once plaintext for an intercepted message has been obtained a decryption-dictionary can be built because the corresponding keystream is known
  - It has modest space-requirements of about 24GB for storing 2^{24} key-streams of perhaps 1500 Bytes in size

BUT AREN'T 40BIT KEYS ANYWAYS VULNERABLE TO BRUTE-FORCE ATTACKS?

- Many manufacturers use 104Bit Keys which are not as vulnerable to brute force attacks
  - But the dictionary size does not depend on the Keysize - only on the IV-size
- Usually many users utilize the same key which is generally not changed too often

Apparently the designers **knew** about the dangers of keystream reuse, but nevertheless **failed** to protect the protocol from the pitfalls that keystream reuse poses.

WHY IS WEP INSECURE? – CONFIDENTIALITY

- IP-Redirection
  - The AP needs to be connected to the Internet which is fairly common
  - Target: Change the IP-Adress of a packet to an IP controlled by the attacker
    - The IP needs to be known
    - The IP-Checksum needs to be modified, too
      - If the IP-Checksum is known, we can simply modify Target-IP and IP-Checksum

IP-REDIRECTION

- If not we need to either decrypt the first packet using another method or
  - We only need to decrypt the first packet because only one field changes in the communication between the same hosts
- Guess the checksum (we have unlimited tries since the AP will simply discard invalid packets)
  - Not all 2^{16} possibilies have the same likelyhood
- Compensate for the change of the Target-IP by for example changing the Source-IP
  - Might result in the packet being dropped due to egress filtering rules
- Arrange that the checksum doesn‘t change
  - If the original destination is 10.20.30.40 and the attacker controls the 192.168.0.0/16 subnet he can simply choose 192.168.103.147

A REACTION ATTACK

- Based on three properties of the TCP-Protocol
  - Packets are only accepted if their checksum is valid
  - An acknowledgement package (TCP ACK) can be easily identified by its size
  - If the flipped bit is chosen cleverly the TCP-Checksum is only undisturbed if the one-bit condition $P_i \oplus P_{i+16} = 1$ holds
    - Thus each request with one bit flipped can reveal one bit of the plaintext
    - By repeating the attack most bits of the message can be deduced

BITTAU'S FRAGMENTATION ATTACK

- A client is able to split a packet into up to 16 fragments; each of them is encrypted separately.
- After an attacker has discovered a single key stream of length m, he can send packets with ((m – 4)*16) = 16*m – 64 bytes of arbitrary payload (length of the ICV excluded) and recover a key stream of length 16*m–60 bytes, by splitting them into up to 16 separate fragments.