

Security Topics in Networking and Distributed Systems

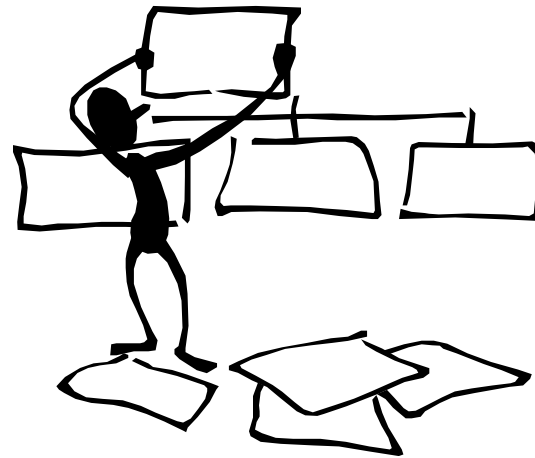
CS 590D

Lecture 14: WEP

Department of Computer Sciences
Purdue University

References

- Intercepting Mobile Communications: The Insecurity of 802.11 Nikita Borisov, Ian Goldberg, David Wagner, INFOCOM 2001.
- Using the Fluhrer, Mantin, and Shamir Attack to Break WEP , Adam Stubblefield, John Ioannidis, and Aviel D. Rubin, NDSS 2002.



Wired Equivalent Privacy

- Security goals: protect link-level transmission
 - Confidentiality
 - Access control
 - Data integrity
- Security relies on the difficulty of discovering the secret key through a brute-force attack
- Uses stream cipher RC4 for encryption and CRC32 for integrity

WEP Details

- RC4 is a stream cipher: based on key k and initialization vector (IV) v , generates a keystream $RC4(v,k)$
- To send a message M from A to B
 - Compute integrity checksum (CRC32): $c(M)$
 - plaintext $P = \{M, c(M)\}$
 - Encrypt P using RC4: ciphertext $C = P \oplus RC4(v,k)$
 - Transmit **$C' = v, (P \oplus RC4(v,k))$**
- To decipher an encrypted message C' , the encryption process is reversed

Some Observations

- The integrity check does not depend on a key, but just on the message M , so anybody can create a pair M and $\text{CRC32}(M)$
- The WEP standard specifies **40 bit keys** and 24 IV, sometimes referred as a 64-bit key. Some vendor implemented 128-bit keys (24 IV and **104 bit key**).
- **The IV is sent in clear, so is available to the attacker as well.**

Risk of Keystream Reuse

$$C1 = P1 \oplus RC4(v, k)$$

$$C2 = P2 \oplus RC4(v, k)$$

$$C1 \oplus C2 = P1 \oplus P2$$

- If P1 or P2 is also known by the attacker, the other plaintext is easy to compute
- If n ciphertexts using the same keystream are available makes reading traffic easier (frequency analysis, etc)
- *Find plaintext P and the encryption C with keystream k, then it is easy to decipher any ciphertext C' encrypted with the same keystream k.*

Is Keystream Reused?

- The pseudorandom keystream is based on the shared key k and the initialization vector IV . Since the key k is secret and is difficult to be changed for every packet, changing the IV is important to prevent keystream reuse.
- The IV is sent in clear, so is available to the attacker as well.
- The WEP standard recommends, but does not require that the IV be changed every packet, but does not say anything about how to select the IV .
- An implementation can reuse the same IV for all packets without risking non-compliance !

24-bit IV Space

- Busy access point sending 1500 byte packets, at an average of 2 Mbps, exhausts the space in **half a day**.
- Random generation of IV can produce collisions every **5000** packets (due to the *birthday paradox*).
- Many implementations use for IV a counter that is incremented for each packet sent and reset every time the card is inserted in the computer.

Exploiting Keystream Reuse

- Methods to obtain pairs (plaintext, ciphertext):
 - IP fields predictable: login sequences, recognize shared libraries transfer
 - Send email and wait for the user to check it via wireless links
 - Send data to access-points that have access control disables and observe the encrypted data

Dictionary Attack

- Goal: Decrypt traffic
- How: Store keystream in a table, indexed by IV.
- Remember the IV is sent in clear
- When the attacker sees a packet with an IV stored already in the table, look up the corresponding keystream, XOR it against the packet, and read the data!
- Table is at most $1500 * 2^{24}$ bytes = 24 GB

Packet Modification

- CRC32 is linear: $c(M \oplus D) = c(M) \oplus c(D)$
- Message M was transmitted, and the ciphertext was C and the IV was IV , C and IV are known to the adversary.
- Attacker can find C' s. t. it decrypts to M' , $M' = M \oplus D$
 $D =$ arbitrarily chosen by the attacker
- $C' = C \oplus \langle D, c(D) \rangle$
 - $= RC4(v, k) \oplus \langle M, c(M) \rangle \oplus \langle D, c(D) \rangle$
 - $= RC4(v, k) \oplus \langle M \oplus D, c(M) \oplus c(D) \rangle$
 - $= RC4(v, k) \oplus \langle M', c(M \oplus D) \rangle$
 - $= RC4(v, k) \oplus \langle M', c(M') \rangle$

Packet Injection

- The attacker knows the keystream, he can select any message and compute CRC of the message **without knowing the key**.
- The base station will accept the packet as valid

WEP Authentication

- *Base station verifies that a client joining the network really knows the shared secret key k .*
- The base station sends a challenge string to the client, and the client sends back the encrypted challenge
- The base station checks if the challenge is correctly encrypted, and if so, accepts the client.
- If adversary sees a challenge/response pair for a given key k ; he can perform the packet injection attack previously describe, and trick the base station.

Fluher, Mantin, and Shamir Attack

- This is an known-plaintext attack against RC4, that allows attackers to eventually recover a key.
- Attack is based on an assumption that the attacker is able to guess the first byte of plaintext used by the victim.
- Stubblefield, Ionnandis, and Rubin showed that the attack is possible in practice

RC4

- A proprietary cipher owned by RSA DSI, designed by Ron Rivest.
- Simple and effective design.
- Variable key size, byte-oriented stream cipher.
- Widely used (web SSL/TLS, wireless WEP).
- Key forms random permutation of all 8-bit values.
- Uses that permutation to scramble input info processed a byte at a time.

RC4 Key Schedule

- Walks each entry in an array S of numbers: $0..255$ turn, using its current value plus the next byte of key to pick another entry in the array, and swaps their values over.
- Total number of possible states is $256!$, very big number
- S forms **internal state** of the cipher, L is the size of the key k

```
for i = 0 to 255 do
    S[i] = i
j = 0
for i = 0 to 255 do
    j = (j + S[i] + k[i mod L]) (mod 256)
    swap (S[i], S[j])
```

RC4 Encryption

- Encryption continues shuffling array values
- Sum of shuffled pair selects the "stream key" byte value
- XOR with next byte of message to en/decrypt

$i = j = 0$

for each message byte m_i

$i = (i + 1) \pmod{256}$

$j = (j + S[i]) \pmod{256}$

swap($S[i]$, $S[j]$)

$t = (S[i] + S[j]) \pmod{256}$

$C_i = m_i \oplus S[t]$

RC4 Cryptanalysis

- The algorithm was kept secret however...
- In 1994 the source code was leaked on the to *cyberpunks* mailing list.
- The external analysis of RC4 was done on the source code that leaked in 1994.
- Fluhrer showed two weaknesses:
 - the first byte generated by RC4 leaks information about individual key bytes.
 - found a large number of weak keys, in which knowledge of a small number of key bits suffices to determine many state and output bits with non-negligible probability.



The Attack

- The first bits of the output are always going to be based on the first values of Sbox since x and y are initialized to zero.
- $x = (x+1) \bmod 256$
- $y = (y+S_x) \bmod 256$
- swap S_x and S_y
- $t = (S_x + S_y) \bmod 256$
- $K = S_t$
- Statistical attack that allows an attacker to recover the key after 60 different IVs and the same key: they estimate 4,000,000 pkts.

Stubblefield, Ionnandis, and Rubin

- Implemented the attack using inexpensive hardware.
- Identified other weaknesses in WEP
 - the keys are ascii, and therefore it limited the possible key space since numbers were based on ascii equivalents to letters.
- WEP is a link layer protocol: it encrypts the network layer data.
 - First byte is going to be the IP packet.
 - Worse, 802.11, in order to be compatible with IP as well as IPX and other network protocols, uses the 802.2 logical link layer encapsulation.
 - This just means that all packets always start with the same 802.2 header.
 - **Guessing the first byte is trivial.**

Countermeasures

- Improve key management: every host should have its own key and key should be changed frequently. **Note that this will not solve the attacks on message authentication.**
- Use higher-level security mechanisms such as IPSec, SSH, and VPN for security, instead of relying on WEP.
- Treat all systems that are connected via 802.11 as external. Place all access points outside the firewall.

Lessons Learnt

- Engineering network protocols vs. security:
 - CRC-32 and RC4 are fast and simple, but they have problems
 - Being stateless and liberal are good for networking, but dangerous for security because they give an attacker more freedom
- Learn from previous works: see IPSEC, TLS.
- Public review is important: international standards should be examined by the cryptographic community

Summary of Attacks on WEP

- Finding the key just by observing the traffic
- Decrypting traffic looking for pairs of plaintext, ciphertext and look for text encrypted with the same keystream
- Packet modification and injection

