Wireless Revolution

Vulnerabilities of WEP

Department of Computer Sciences
Purdue University
Outline

- Security services
- 802.11
- WEP
- Attacks on WEP without finding the key
- Breaking the key
- Countermeasures
- Lessons learnt
Additional Reading

  Nikita Borisov, Ian Goldberg, David Wagner

- * Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. 
  Adam Stubblefield, John Ioannidis, Aviel D. Rubin
Security Services

• **1) Confidentiality:** information is available for reading only to authorized parties.

  **Example:** Alice sends a message to Bob, only Alice and Bob can understand the content of the message.

• **2) Authentication:**
  - Data source authentication: the data is coming from an authorized party.
    **Example:** Alice receives a message from Bob. This service ensures that the message is from Bob and not from Carl.
  - Entity authentication: the entity is who it says it is.
    **Example:** When Alice tries to obtain access to her bank account, an authentication operation is performed to ensure that Alice asks for the information.
Security Services (2)

- **3) Data integrity**: detect if data was modified, from the source to the destination.
  
  **Example**: Alice sends an email to Bob. Carl intercepts the message and modifies it. Data integrity allows for Bob to detect that the message was modified on the way from Alice to him.

- **4) Non-repudiation**: neither the sender, nor the receiver of a message are able to deny the transmission.
  
  **Example**: Alice sends Bob a contract, signed. The non-repudiation service ensures that Alice can not claim that the signature was produced by somebody else.
Security Services (3)

- **5) Access control**: only authorized parties can use specific resources.
  
  **Example**: Alice wants to print a document, she must be authorized to get that document and to use the printer.

- **6) Availability**: resources available to authorized parties.
  
  **Example**: A web site might become unavailable if the server crashes, or is bombarded with requests.
Security Attacks

- **Passive:** the attacker does not modify the data, only monitors the communication. It threatens confidentiality.
  
  Example: listen to the communication between Alice and Bob, and if it’s encrypted try to decrypt it.

- **Active:** the attacker is actively involved in deleting, adding or modifying data. It threatens data integrity, authentication and confidentiality.
  
  Example: Alice sends Bob a message: ‘meet me today at 5’, Carl intercepts the message and modifies it ‘meet me tomorrow at 5’, and then sends it to Bob.
Security Attacks: Examples

- Interruption
- Interception
Security Attacks: Examples

- Modification
- Fabrication (injection)
Security Mechanisms

- **Cryptography**: for example encrypt data.
- **Software**: access limitations in a database, in operating system protect each user from other users, networking: firewall.
- **Hardware**: use smartcard for authentication.
- **Policies**: define who has access to what resources, frequent changes of passwords, etc.
- **Physical Security**: place the device in a locked room
802.11

- **802.11b**: IEEE specification for 11 Mbit/s Ethernet (CSMA/CD) running in 2.4 GHz bandwidth space
- **802.11a**: IEEE specification for 54 Mbit/s Ethernet (CSMA/CD) running in 5 GHz bandwidth space
- Modes of operation: infrastructure and ad hoc
Operation Modes

infrastructure

ad hoc
What’s the big deal about security?

Wireless communication is radio communication, anybody can listen or inject traffic
Wired Equivalent Privacy

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

- RC4 is a **stream cipher**: based on secret key $k$ and initialization vector (IV) $v$, generates a keystream $RC4(IV,k)$
- 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 $C = P \oplus RC4(IV,k)$
  - Transmit $C' = IV, (P \oplus RC4(IV,k))$
- 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 $c(M)$

• In the 64-bit WEP standard, a 64-bit key = 40 bit key and 24 IV, the key is actually just 40 bits (128-bit key = 24 IV and 104 bit key).

• The IV is sent in clear, so is available to the attacker as well.

• The key does not change
Risk of Keystream Reuse

\[ C_1 = P_1 \oplus \text{RC4(IV, k)} \]
\[ C_2 = P_2 \oplus \text{RC4(IV, k)} \]
\[ C_1 \oplus C_2 = P_1 \oplus P_2 \]

- 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)
- **Attack:** 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, also 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 IV 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 in Practice

• 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 disabled and observe the encrypted data
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.
• Attack: Attacker can find \( C' \) s. t. it decrypts to \( M', M' = M \oplus D \), D is arbitrarily chosen by the attacker

\[
C' = C \oplus <D,c(D)>
\]

\[
= RC4(IV,k) \oplus <M,c(M)> \oplus <D,c(D)>
\]

\[
= RC4(IV,k) \oplus <M \oplus D, c(M) \oplus c(D)>
\]

\[
= RC4(IV,k) \oplus <M', c(M \oplus D)>
\]

\[
= RC4(IV,k) \oplus <M', c(M')>
\]
Packet Injection

- The attacker knows the keystream, he can select any message and compute CRC of the message without knowing the key.
- The access point will accept the packet as valid
WEP Authentication

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

• Security services
• 802.11
• WEP
• Attacks on WEP without finding the key
• Breaking the key
• Countermeasures
• Lessons learnt
Fluher, Mantin, and Shamir Attack

- This is an known-plaintext attack against RC4, that allows attackers to eventually recover the 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 $k$

```java
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

\[
i = j = 0
\]
\[
\text{for each message byte } m_i
\]
\[
i = (i + 1) \mod 256
\]
\[
j = (j + S[i]) \mod 256
\]
\[
\text{swap}(S[i], S[j])
\]
\[
t = (S[i] + S[j]) \mod 256
\]
\[
C_i = m_i \oplus S[t]
\]
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.
- \( x = (x+1) \mod 256 \)
- \( y = (y+S_x) \mod 256 \)
- swap \( S_x \) and \( S_y \)
- \( t = (S_x + S_y) \mod 256 \)
- \( K = S_t \)
- 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 the 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

- The strongest attack is finding the key just by observing the traffic and exploiting a known-attack on RC4, the encryption algorithm.
- Decrypting traffic looking for pairs of plaintext, ciphertext and look for text encrypted with the same keystream.
- Injecting traffic by exploiting the fact that integrity was implemented using CRC32.