Network Security

Chapter 15

Security of Wireless Local Area Networks
IEEE 802.11

- IEEE 802.11 standardizes medium access control (MAC) and physical characteristics of a wireless local area network (LAN)

- The standard comprises three physical layer units:
  - Frequency Hop Spread Spectrum: 2.4 GHz band; 1, 2, 5.5, and 11 Mbit/s
  - Direct Sequence Spread Spectrum: 2.4 GHz band, 1, 2, 5.5, 11, and 22 Mbit/s
  - Baseband infrared: diffuse infrared; 1 and 2 Mbit/s

- Transmission in the license-free 2.4 GHz band implies:
  - Medium sharing with un-volunteering 802.11 devices
  - Overlapping of logical separated wireless LANs
  - Overlapping with non-802.11 devices

- The medium access control (MAC) supports operation under control of an access point as well as between independent stations

- In this class we will mainly focus on the standard’s security aspects:
  - Many equipment vendors claim that IEEE 802.11 is as secure as a wired network (more on this below...)
802.11 - Architecture of an Infrastructure Network

- **Station (STA):**
  - Terminal with access mechanisms to the wireless medium and radio contact to the access point

- **Basic Service Set (BSS):**
  - Group of stations using the same radio frequency

- **Access Point:**
  - Station integrated into the wireless LAN and the distribution system

- **Portal:**
  - Bridge to other (wired) networks

- **Distribution System:**
  - Interconnection network to form one logical network (extended service set, ESS) based on several BSS
802.11 - Architecture of an Ad-Hoc Network

- **Station (STA):**
  - Terminal with access mechanisms to the wireless medium

- **Basic Service Set (BSS):**
  - Group of stations using the same radio frequency

- Ad-Hoc networks allow direct communication between end systems within a limited range

- As there is no infrastructure, no communication is possible between different BSSs
Security Services of IEEE 802.11

- Security services of IEEE 802.11 are realized by:
  - Entity authentication service
  - *Wired Equivalent Privacy (WEP)* mechanism

- WEP is supposed to provide the following security services:
  - Confidentiality
  - Data origin authentication / data integrity
  - Access control in conjunction with layer management

- WEP makes use of the following algorithms:
  - The RC4 stream cipher (please refer to chapter 3)
  - The Cyclic Redundancy Code (CRC) checksum for detecting errors
The Cyclic Redundancy Code (1)

- The cyclic redundancy code (CRC) is an error detection code.
- Mathematical basis:
  - Treat bit strings as representations of polynomials with coefficients 0 and 1 ⇒ a bit string representing message $M$ is interpreted as $M(x)$.
  - Polynomial arithmetic is performed modulo 2 ⇒ addition and subtraction are identical to XOR.
- CRC computation for a message $M(x)$:
  - A and B agree upon a polynomial $G(x)$; usually $G(x)$ is standardized.
  - Let the $n$ be the degree of $G(x)$, that is the length of $G(x)$ is $n + 1$.
  - Then if $\frac{M(x) \times 2^n}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$ it holds $\frac{M(x) \times 2^n + R(x)}{G(x)} = Q(x)$.
    where $R(x)$ is the remainder of $M(x)$ divided by $G(x)$.
  - Usually, $R(x)$ is appended to $M(x)$ before transmission and $Q(x)$ is not of interest, as it is only checked if $\frac{M(x) \times 2^n + R(x)}{G(x)}$ divides with remainder 0.
The Cyclic Redundancy Code (2)

- Consider now two Messages $M_1$ and $M_2$ with CRCs $R_1$ and $R_2$:
  - As $\frac{M_1(x) \times 2^n + R_1(x)}{G(x)}$ and $\frac{M_2(x) \times 2^n + R_2(x)}{G(x)}$ divide with remainder 0
  - Also $\frac{M_1(x) \times 2^n + R_1(x) + M_2(x) \times 2^n + R_2(x)}{G(x)} = \frac{(M_1(x) + M_2(x)) \times 2^n + (R_1(x) + R_2(x))}{G(x)}$ divides with remainder 0

  $\Rightarrow$ CRC is linear, that is $\text{CRC}(M_1 + M_2) = \text{CRC}(M_1) + \text{CRC}(M_2)$

- This property renders CRC weak for cryptographic purposes! (more on this below...)

IEEE 802.11 Entity Authentication (1)

- IEEE 802.11 authentication comes in two “flavors”:
  - **Open System Authentication:**
    - “Essentially it is a null authentication algorithm.” (IEEE 802.11, section 8.1.1)
  - **Shared Key Authentication:**
    - “Shared key authentication supports authentication of STAs as either a member of those who know a shared secret key or a member of those who do not.” (IEEE 802.11, section 8.1.2)
    - “The required secret, shared key is presumed to have been delivered to participating STAs via a secure channel that is independent of IEEE 802.11”
IEEE 802.11 Entity Authentication (2)

- IEEE 802.11’s *Shared Key Authentication* dialogue:
  - Authentication should be performed between stations and access points and could also be performed between arbitrary stations.
  - When performing authentication, one station is acting as the *requestor* (A) and the other one as the *responder* (B).
  - The authentication dialogue:
    1. \( A \rightarrow B: (\text{Authentication}, 1, \text{ID}_A) \)
    2. \( B \rightarrow A: (\text{Authentication}, 2, r_B) \)
    3. \( A \rightarrow B: \{\text{Authentication}, 3, r^*_B\}^ {K_{A,B}} \)
    4. \( B \rightarrow A: (\text{Authentication}, 4, \text{Successful}) \)

Mutual authentication requires two independent protocol runs, one in each direction.

- But: an attacker can impersonate after eavesdropping one protocol run, as he can obtain a valid keystream from messages 2 and 3!
IEEE 802.11’s Wired Equivalence Privacy (1)

- IEEE 802.11’s WEP uses RC4 as a pseudo-random-bit-generator (PRNG):
  - For every message $M$ to be protected a 24 bit *initialization vector (IV)* is concatenated with the shared key $K_{BSS}$ to form the seed of the PRNG
  - The *integrity check value (ICV)* of $M$ is computed with CRC and appended (“$||$”) to the message
  - The resulting message $(M || ICV)$ is XORed (“⊕”) with the keystream generated by $RC4(IV || K_{BSS})$
IEEE 802.11’s Wired Equivalence Privacy (2)

- As IV is sent in clear with every message, every receiver who knows $K_{BSS}$ can produce the appropriate keystream to decrypt a message.
  - This assures the important *self-synchronization property* of WEP.

- The decryption process is basically the inverse of encryption:

![WEP Decryption Block Diagram](image)
IEEE 802.11’s Security Claims

- The WEP has been designed to ensure the following security properties:
  - Confidentiality:
    - Only stations which possess $K_{BSS}$ can read messages protected with WEP
  - Data origin authentication / data integrity:
    - Malicious modifications of WEP protected messages can be detected
  - Access control in conjunction with layer management:
    - If set so in the layer management, only WEP protected messages will be accepted by receivers
    - Thus stations that do not know $K_{BSS}$ can not send to such receivers

- Unfortunately, none of the above claims holds... :o(
Weakness #1: The Keys

- IEEE 802.11 does not specify any key management:
  - Manual management is error prone and insecure
  - Shared use of one key for all stations of a BSS introduces additional security problems
  - As a consequence of manual key management, keys are rarely changed
  - As another consequence, “security” is often even switched off!

- Key Length:
  - The key length of 40 bit specified in the original standard provides only poor security
  - The reason for this was exportability
  - However, today’s wireless LAN cards often also allow keys of length 128 bit
Weakness #2: WEP Confidentiality is Insecure

- Even with well distributed and long keys WEP is insecure.
- The reason for this is reuse of keystream:
  - Recall that encryption is re-synchronized with every message by pre-pending an IV of length 24 bit to $K_{BSS}$ and re-initializing the PRNG.
  - Consider two plaintexts $M_1$ and $M_2$ encrypted using the same IV$_1$:
    - $C_1 = P_1 \oplus \text{RC4}(\text{IV}_1, K_{BSS})$
    - $C_2 = P_2 \oplus \text{RC4}(\text{IV}_1, K_{BSS})$
    - then:
      - $C_1 \oplus C_2 = (P_1 \oplus \text{RC4}(\text{IV}_1, K_{BSS})) \oplus (P_2 \oplus \text{RC4}(\text{IV}_1, K_{BSS})) = P_1 \oplus P_2$
  - Thus, if an attacker knows, for example, $P_1$ and $C_1$ he can recover $P_2$ from $C_2$ without knowledge of the key $K_{BSS}$.
  - Cryptographers call this an attack with known-plaintext.
- How often does reuse of keystream occur?
  - In practice quite often, as many implementations choose IV poorly.
  - Even with optimum choice, as IV’s length is 24 bit, a busy base station of a 11 Mbit/s WLAN will exhaust the available space in half a day.
Weakness #3: WEP Data Integrity is Insecure

- Recall that CRC is a linear function and RC4 is linear as well
- Consider A sending an encrypted message to B which is intercepted by an attacker E:
  - $A \rightarrow B: (IV, C)$ with $C = \text{RC4}(IV, K_{BSS}) \oplus (M, \text{CRC}(M))$
- The attacker E can construct a new ciphertext $C'$ that will decrypt to a message $M'$ with a valid checksum $\text{CRC}(M')$:
  - E chooses an arbitrary message $\Delta$ of the same length
  - $C' = C \oplus (\Delta, \text{CRC}(\Delta)) = \text{RC4}(IV, K_{BSS}) \oplus (M, \text{CRC}(M)) \oplus (\Delta, \text{CRC}(\Delta))$
    - $= \text{RC4}(IV, K_{BSS}) \oplus (M \oplus \Delta, \text{CRC}(M) \oplus \text{CRC}(\Delta))$
    - $= \text{RC4}(IV, K_{BSS}) \oplus (M \oplus \Delta, \text{CRC}(M \oplus \Delta))$
    - $= \text{RC4}(IV, K_{BSS}) \oplus (M', \text{CRC}(M'))$
- Note, that E does not know $M'$ as it does not know $M$
- Nevertheless, a “1” at position $n$ in $\Delta$ results in a flipped bit at position $n$ in $M'$, so E can make controlled changes to $M$
  - $\Rightarrow$ Data origin authentication / data integrity of WEP is insecure!
Weakness #4: WEP Access Control is Insecure

- Recall that the integrity function is computed without any key.
- Consider an attacker who learns a plaintext-ciphertext pair:
  - As the attacker knows M and C = RC4(IV, K_{BSS}) \oplus (M, CRC(M)), he can compute the keystream used to produce C.
  - If E later on wants to send a message M' he can compute C' = RC4(IV, K_{BSS}) \oplus (M', CRC(M')) and send the message (IV, C')
  - As the reuse of old IV values is possible without triggering any alarms at the receiver, this constitutes a valid message.
  - An “application” for this attack is unauthorized use of network resources:
    - The attacker sends IP packets destined for the Internet to the access point which routes them accordingly, giving free Internet access to the attacker.

⇒ WEP Access Control can be circumvented with known plaintext.
Weakness #5: Weakness in RC4 Key Scheduling

- In early August 2001 a new attack to WEP was discovered:
  - The shared key can be retrieved in less than 15 minutes provided that about 4 to 6 million packets have been recovered.
  - The attack is basically a known-plaintext attack, that makes use of the following properties of RC4 and WEP’s usage of RC4:
    - RC4 is vulnerable to deducing bits of a key if:
      - many messages are encrypted with keystream generated from a variable initialization vector and a fixed key, and
      - the initialization vectors and the plaintext of the first two octets are known for the encrypted messages.
    - The IV for the keystream is transmitted in clear with every packet.
    - The first two octets of an encrypted data packet can be guessed.
  - The attack is described in [SMF01a] and [SIR01a].
  - R. Rivest comments on this [Riv01a]:
    “Those who are using the RC4-based WEP or WEP2 protocols to provide confidentiality of their 802.11 communications should consider these protocols to be broken [...].”
Conclusions on IEEE 802.11’s Deficiencies

- IEEE 802.11 does not provide sufficient security:
  - Missing key management makes use of the security mechanisms tedious and leads to rarely changed keys or even security switched off
  - Entity authentication as well as encryption rely on a key shared by all stations of a basic service set
  - Insecure entity authentication protocol
  - Reuse of keystream makes known-plaintext attacks possible
  - Linear integrity function allows to forge ICVs
  - Unkeyed integrity function allows to circumvent access control by creating valid messages from a known plaintext-ciphertext pair
  - Weakness in RC4 key scheduling allows to cryptanalyze keys
- Even with IEEE 802.1x and individual keys the protocol remains weak
- Some proposed countermeasures:
  - Place your IEEE 802.11 network outside your Internet firewall
  - Do not trust any host connected via IEEE 802.11
  - Additionally, use other security protocols, e.g. PPTP, L2TP, IPSec, SSH, ...
Fixing WLAN Security: IEEE 802.11 Task Group i

- **Scope:**
  - Enhance the 802.11 Medium Access
  - Defining the interaction between 802.1X and 802.11 standards

- **TGii defines two classes of security algorithms for 802.11:**
  - Pre-RSN security Network (→ WEP)
  - Robust Security Network (RSN)

- **RSN security consists of two basic subsystems:**
  - **Data privacy mechanisms:**
    - TKIP - rapid re-keying to patch WEP for minimum privacy
    - AES encryption - robust data privacy for long term (WRAP, CCMP)
  - **Security association management:**
    - 802.1X authentication - replacing 802.11 authentication
    - 802.1X key management - to provide cryptographic keys

*(most material on 802.11i is taken from [WM02a]*)
The Long Term Solution: AES based WLAN Protection

- For “political” reasons, two protocols have been defined:
  - CCMP (AES in a combined counter mode with CBC-MAC)
  - WRAP (will be only optional due to IPR issues)

- CCMP:
  - Mandatory to implement: the long-term solution
  - An all new protocol with few concessions to WEP
  - Provides: data confidentiality, data origin authentication, replay protection
  - Based on AES in Counter Mode Encryption with CBC-MAC (CCM)
    - Use CBC-MAC to compute a MIC on the plaintext header, length of the plaintext header, and the payload
    - Use CTR mode to encrypt the payload with counter values 1, 2, 3, …
    - Use CTR mode to encrypt the MIC with counter value 0

- AES overhead requires new AP hardware
- AES overhead may require new STA hardware for hand-held devices, but not PCs (however, this will increase CPU load and energy consumption)
An Intermediate Solution: Temporal Key Integrity Protocol

- **Design Goals:**
  - Quick fix to the existing WEP problem, runs WEP as a sub-component
  - Can be implemented in software, reuses existing WEP hardware
  - Requirements on existing AP hardware:
    - 33 or 25 MHz ARM7 or i486 already running at 90% CPU utilization before TKIP
    - Software/firmware upgrade only
    - Don’t unduly degrade performance

- **Main concepts:**
  - Message Integrity Code (MIC)
  - Countermeasures in case of MIC failures
  - Sequence counter
  - Dynamic key management (re-keying)
  - Key mixing

- TKIP meets criteria for a good standard: everyone is unhappy with it...
TKIP MPDU Data Format

IV / KeyID
4 octets

Extended IV
4 octets

Data \geq 1 octets

MIC
8 octets

ICV
4 octets

RC4Key
[0]

RC4Key
[1]

RC4Key
[2]

Rsvd

Ext IV

Key ID

TSC2

TSC3

TSC4

TSC5
TKIP Design: Message Integrity Code Function Michael

- Protect against forgeries:
  - Must be cheap: CPU budget 5 instructions / byte
  - Unfortunately is weak: a $2^{29}$ message attack exists
  - Computed over MSDUs, while WEP is over MPDUs
  - Uses two 64-bit keys, one in each link direction
  - Requires countermeasures:
    - Rekey on active attack (only few false alarms as CRC is checked first)
    - Rate limit rekeying to one per minute

![Diagram of TKIP Design](image)
TKIP Design: Replay Protection and RC4 Key Scheduling

- **Replay protection:**
  - Reset packet sequence # to 0 on rekey
  - Increment sequence # by 1 on each packet
  - Drop any packet received out of sequence

- **Circumvent WEP’s encryption weaknesses:**
  - Build a better per-packet encryption key by preventing weak-key attacks and decorrelating WEP IV and per-packet key
  - must be efficient on existing hardware

```
Transmit Address: 00-A0-C9-BA-4D-5F
Packet Sequence #
```

```
Base key
```

```
4 msb
```

```
2 lsb
```

```
Phase 1 Mixer
```

```
Phase 2 Mixer
```

```
Intermediate key
```

```
Per-packet key
```

```
TKIP Design: Replay Protection and RC4 Key Scheduling
```
TKIP Processing at the Sender

Temporal Key

Phase 1 key mixing

TTAK Key

MAC Key

SA + DA + Plaintext MSDU Data

MIC Key

TKIP sequence counter(s)

MIC

Plaintext MSDU + MIC

Fragment(s)

Phase 2 key mixing

WEP seed(s) (represented as WEP IV + RC4 key)

WEP Encapsulation

Ciphertext MPDU(s)

(source: IEEE 802.11 Tgi draft)
TKIP Processing at the Receiver

Phase 1 key mixing

Temporal Key
TA

MIC Key
TTAK Key

TKIP sequence counter
Unmix IV

Ciphertext MPDU
WEP IV

Key mixing

In-sequence MPDU

Unmix IV

Out-of-sequence MPDU

WEP Decapsulation

Plaintext MPDU

Reassemble

SA + DA + Plaintext MSDU

MIC

MIC’

MIC = MIC’?

Plaintext MSDU

Countermeasures

(source: IEEE 802.11 Tgi draft)
## Comparison of WEP, TKIP, and CCMP

<table>
<thead>
<tr>
<th></th>
<th>WEP</th>
<th>TKIP</th>
<th>CCMP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cipher</strong></td>
<td>RC4</td>
<td>RC4</td>
<td>AES</td>
</tr>
<tr>
<td><strong>Key Size</strong></td>
<td>40 or 104 bits</td>
<td>128 bits</td>
<td>128 bits encrypt, 64 bit auth.</td>
</tr>
<tr>
<td><strong>Key Life</strong></td>
<td>24-bit IV, wrap</td>
<td>48-bit IV</td>
<td>48-bit IV</td>
</tr>
<tr>
<td><strong>Packet Key</strong></td>
<td>Concat.</td>
<td>Mixing Fnc.</td>
<td>Not Needed</td>
</tr>
<tr>
<td><strong>Integrity</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>CRC-32</td>
<td>Michael</td>
<td>CCM</td>
</tr>
<tr>
<td><strong>Header</strong></td>
<td>None</td>
<td>Michael</td>
<td>CCM</td>
</tr>
<tr>
<td><strong>Replay</strong></td>
<td>None</td>
<td>Use IV</td>
<td>Use IV</td>
</tr>
<tr>
<td><strong>Key Mgmt.</strong></td>
<td>None</td>
<td>EAP-based</td>
<td>EAP-based</td>
</tr>
</tbody>
</table>

At least for the near future TKIP is most likely to provide basic WLAN security.
Additional References


