

WLAN Security

AES and CCMP

(C) Herbert Haas 2005/03/11

Content

In this chapter a detailed overview about today's WLAN security problems and solutions are presented.

This subchapter provides an introduction into AES and CCMP.

Objective

After completing this chapter the following tasks could be solved:

- Emphasize the basic vulnerabilities of WLAN
- Explain why WEP is insecure and give a mathematical example
- Compare WEP and TKIP/MIC
- Highlight the design flaws of the WLAN standard authentication
- Explain the design idea of 802.1x
- Compare EAP-TLS, LEAP, PEAP, EAP-TTLS, EAP-FAST with each other and emphasize important security features
- Explain the design concept of WPA and WPA2
- Implement a reliable 802.1x infrastructure over a WAN connection
- List important issues to be considered when choosing a VPN design
- Explain PSPF

802.11i



Pre-standard
802.11i – TSN
(WPA)

- **Message Integrity Check (MIC)**
 - ◆ Nonlinear algorithm
- **Temporal Key Integrity Protocol (TKIP or “WEP2”)**
 - ◆ Also uses RC4-based WEP without the known flaws
 - Per-packet keys through IV mixing
 - Replay protection
 - ◆ Essentially a patch for WEP

Ratified 802.11i
– RSN
(WPA2)

First WPA2 certifications
already since 1st Sept 2004

- **Counter Mode CBC MAC (CCMP)**
 - ◆ = AES + CBC-MAC
 - ◆ Replaces WEP !!!
(requires new HW support)

Recently, the **IEEE 802.11i** Security Task Group released two "informative texts" providing WEP hardening: MIC and TKIP. The IEEE 802.11 Task Group "i" is working on standardizing WLAN encryption improvements. Two new network types, called Transition Security Network (TSN) and Robust Security Network (RSN) had been defined.

The Temporal Key Integrity Protocol (TKIP), initially referred to as WEP2) is an interim solution (as part of TSN) that fixes the key reuse problem of WEP.

TKIP is a compromise on strong security and possibility to use existing hardware. Still uses RC4 but per-packet keys plus replay protection through a keyed packet authentication mechanism (Michael MIC).

TKIP begins with a 128 bit "temporal key" shared among clients and access points. TKIP combines the temporal key with the client's MAC address and then adds a 6-byte IV to produce the key that will encrypt the data. Thus each station uses different key streams for encryption. TKIP changes keys every 10,000 packets, using a dynamic distribution method.

The IEEE plans to use the **Advanced Encryption Standard (AES)** instead of RC4 for TKIP in the long run (RSN), combined with Counter Mode - Cipher Block Chaining - Message Authentication Code (CBC MAC) to provide strong integrity and message authentication. Also the term "Wireless Robust Authenticated Protocol" (WRAP) is sometimes used synonymously for this concept.

The **Wi-Fi** specified TKIP and MIC as mandatory features of the **Wi-Fi Protected Access (WPA)** protocol, while AES should be part of WPA2.

Note: WiFi and particular vendors uses **different** TKIP/MIC algorithms, which are not compatible. Even WPA was intended to be an intermediate solution because the WiFi only picked a subset of the IEEE 802.11i working draft 3.0).

WPA2 aka 802.11i



- **Exactly the same as WPA1 except...**
 - ♦ CCMP (AES in counter mode) instead of RC4
 - ♦ HMAC-SHA1 instead of HMAC-MD5 for the EAPoL MIC
- **Against rumors WPA2 is only a LITTLE better than WPA1**
 - ♦ But neither will be cracked in the near future !!!

802.11i: CCMP – Overview



- **AES for data encryption (privacy)**
 - ♦ 128-bit block cipher
 - ♦ No per-packet keying needed
 - ♦ HW-realization recommended
 - ♦ Key-life determined by 48-bit IV
- **AES requires a **feedback mode****
 - ♦ To avoid the risks associated with the trivial Electronic Codebook (ECB) mode
 - Repeating patterns are not hidden
 - Not recommended for messages longer than one block !
- **The IEEE is still deciding which feedback mode to standardize for AES encryption – two choices:**
 - ♦ **Counter Mode CBC MAC (CCM)**
 - Provides encryption, authenticity and integrity
 - Applied on both header and data
 - IV also used to prevent replay attacks
 - WLAN's current favourite
 - ♦ **Offline Code Book (OCB) mode**
 - Problem: patented
 - Also supported by some WLAN vendors

The **802.11i** standard was finished in May 2004 and **approved in June 2004**. The main result, WPA2, includes support for more robust encryption algorithm (CCMP: AES in Counter mode with CBC-MAC) to replace TKIP and **optimizations for handoff** (reduced number of messages in initial key handshake, pre-authentication, and PMKSA caching).

The **Advanced Encryption Standard (AES)** is considered as state-of-the-art encryption method, designed recently, using Rijndael as algorithm and is official successor of DES or 3DES. This 128-bit block cipher is considered unbreakable for the next ten years or so.

How secure is AES compared to RC4? RC4 uses up to 128 bits key length, AES uses 256 bits, that is the AES key is 128 bits longer. If only brute force attacks are assumed (algorithms are save enough) and considering Moore's law (computing power doubles every 18 month), then AES is at least $\log_2(128) \cdot 18$ months ahead, that is more than 10 years, compared to RC4.

CCM is a actually the block cipher *mode* of AES that provides both encryption and authentication. It is a combination of counter-mode encryption and CBC-MAC authentication which are two modes that have been studied extensively for many years. CCM was developed as a non-patented alternative to OCB ("Offset Codebook") for use in secure wireless networks, but it can be used in almost any situation that requires secure communications. With CCM encryption and authentication

Links:

Rijndael description and algorithm:
<http://csrc.nist.gov/CryptoToolkit/aes/rijndael/>

AES Lounge:
<http://www.iaik.tu-graz.ac.at/research/krypto/AES/>

Cipher Block Chaining (CBC)



- **No patent**
- **Encryption and MAC use different nonces**
 - ♦ Collision attacks possible but sufficient mitigation when key management provides frequent key changes
- **Identical ciphertext blocks result only when:**
 - ♦ Same key and
 - ♦ Same plaintext and
 - ♦ Same IV is used
- **CBC is self-synchronizing**
 - ♦ If an error (including loss of one or more entire blocks) occurs in block c_j but not c_{j+1} , then c_{j+2} is correctly decrypted to x_{j+2} .

1. Encryption: $c_0 \leftarrow IV$. For $1 \leq j \leq t$, $c_j \leftarrow E_K(c_{j-1} \oplus x_j)$.
2. Decryption: $c_0 \leftarrow IV$. For $1 \leq j \leq t$, $x_j \leftarrow c_{j-1} \oplus E_K^{-1}(c_j)$.

Although CBC mode decryption recovers from errors in ciphertext blocks, modifications to a plaintext block x_j during encryption alter all subsequent ciphertext blocks. This impacts the usability of chaining modes for applications requiring random read/write access to encrypted data.

An exposed IV might allow a man-in-the-middle (MITM) to change the IV value in-transit. Changing the IV changes only the deciphered plaintext for the first block, without garbling the second block. Any or all bits of the first block plaintext can be changed systematically with complete control.

The most obvious way to prevent deliberate MITM changes to the first block plaintext with the IV is to encipher the IV; that prevents an opponent from changing plaintext bits systematically.

Offset Code Book (OCB)



- **Patented**
- **Combines authentication and encryption**
 - ◆ Slightly faster than CBC encryption
 - ◆ More prone to collision attacks than CBC-MAC
- **If a particular collision on 128-bit values occurs, then an attacker can modify the message without being detected by the OCB authentication function**
 - ◆ Weak authentication algorithm – uses same nonce for encryption and authentication
 - ◆ In order to limit the probability of a successful forgery attempt to less than 2^{-64} change the key after 2^{32} blocks of data
 - ◆ Indeed strong enough for many people but does not justify 128-bit AES as successor of DES

AES-OCB is a mode that operates by augmenting the normal encryption process by incorporating an offset value.

The routine is initiated with a unique nonce (the nonce is a 128-bit number) used to generate an initial offset value. The nonce has the XOR function performed with a 128-bit string (referred to as value L).

The output of the XOR is AES-encrypted with the AES key, and the result is the offset value.

The plain-text data has the XOR function performed with the offset and is then AES-encrypted with the same AES key.

The output then has the XOR function performed with the offset once again. The result is the cipher-text block to be transmitted.

The offset value changes after processing each block by having the XOR function performed on the offset with a new value of L.

See <http://www.cs.ucdavis.edu/~rogaway/ocb/index.html>

OCB Algorithm



Convention: Message M, Key K, Nonce N

Define $L := E_K(0)$ from which the offset $Z_i := \gamma_i \cdot L \oplus R$ follows.
 $R := E_K(N \oplus L)$

Then the message is split into M_1, \dots, M_m , where only M_m is typically a non-128 bit block. The messages M_1, \dots, M_{m-1} are encrypted as follows:

$$\begin{aligned} X_i &:= M_i \oplus Z_i \\ Y_i &:= E_K(X_i) \\ C_i &:= Y_i \oplus Z_i \end{aligned}$$

While M_m is encrypted using μ denoting the length of this block:

$$\begin{aligned} X_m &:= \mu \oplus x^{-1} \cdot L \oplus Z_m \\ Y_m &:= E_K(X_m) \\ C_m &:= M_m \oplus \text{first-}\mu\text{-bits}(Y_m) \end{aligned}$$

The authentication is performed in two steps:

$$\begin{aligned} S &:= M_1 \oplus \dots \oplus M_{m-1} \oplus C_m 0^* \oplus Y_m \\ T &:= \text{first-}\tau\text{-bits}(E_K(S) \oplus Z_m) \end{aligned}$$

$C_m 0^*$... last ciphertext block padded with zeros to full 128 bit length

... "Checksum"

... "MAC Tag" of arbitrary length, depending on security vs. transmission cost trade-off. Typically 32..80 (documentation)