Network Security

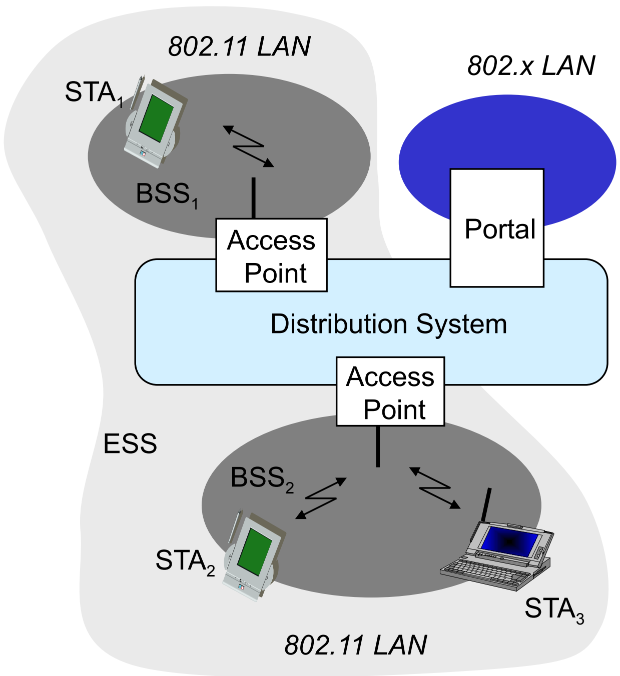
Chapter 15

Security of Wireless Local Area Networks

IEEE 802.11

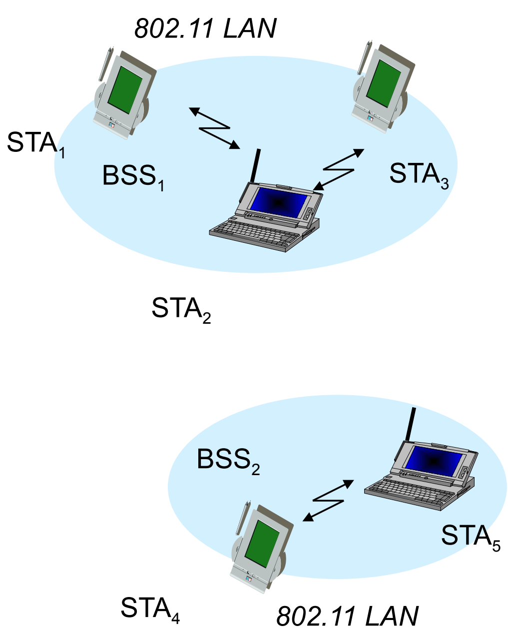
- IEEE 802.11 [IEEE12] standardizes medium access control (MAC) and physical characteristics of a wireless *local area network (LAN)*
- The standard comprises multiple physical layer units:
  - Currently between 1-300 Mbit/s
  - 2.4 GHz band and 5GHz band
  - Many different modulation schemes
- Transmission 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 (in)security aspects!

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 was originally 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

    ⇒ CRC is linear, that is $CRC(M_1 + M_2) = CRC(M_1) + CRC(M_2)$
- This property renders CRC weak for cryptographic purposes! (more on this below...)

IEEE 802.11 Entity Authentication (1)

- Originally IEEE 802.11 authentication come 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.) $A \rightarrow B$: (Authentication, 1, ID_A)

    2.) $B \rightarrow A$: (Authentication, 2, r_B)

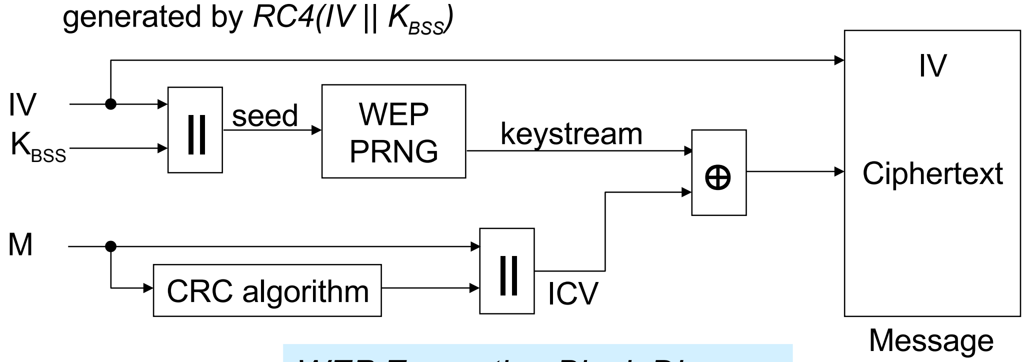
    3.) $A \rightarrow B$: $\{$Authentication, 3, $r_B\}_{K_{A,B}}$

    4.) $B \rightarrow A$: (Authentication, 4, 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)

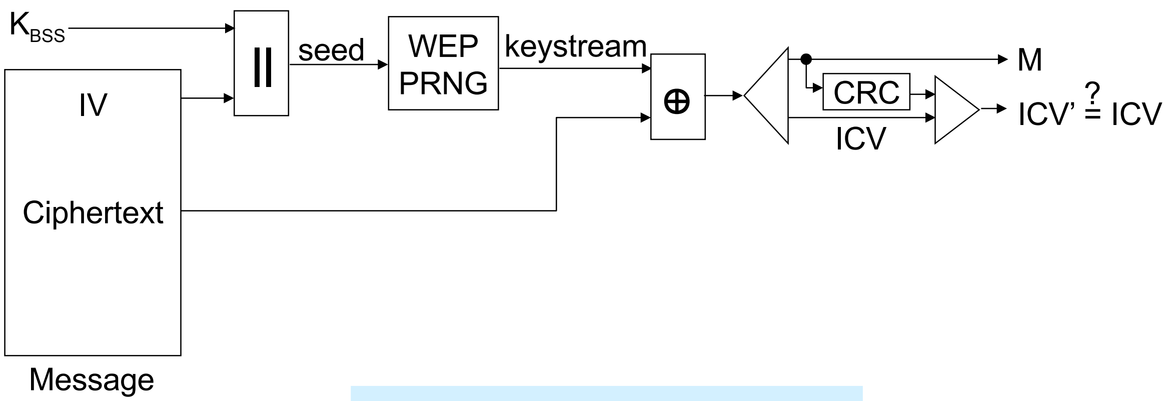
- IEEE 802.11's WEP uses RC4 as a pseudo-random-bit-generator (PRNG):
  - For every message *M* to be protected a 24 bit *initialization vector (IV)* is concatenated with the shared key K_{BSS} to form the seed of the PRNG
  - The *integrity check value (ICV)* of *M* is computed with CRC and appended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



WEP Encryption Block Diagram

IEEE 802.11's Wired Equivalence Privacy (2)

- As *IV* is send 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

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 a 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
  - Wireless LAN cards often also allow keys of length 104 bit, but that does not make the situation better as we will see later

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:
    - $C_1 = P_1 \oplus \text{RC4}(IV_1, K_{BSS})$
    - $C_2 = P_2 \oplus \text{RC4}(IV_1, K_{BSS})$

    then:
    - $C_1 \oplus C_2 = (P_1 \oplus \text{RC4}(IV_1, K_{BSS})) \oplus (P_2 \oplus \text{RC4}(IV_1, K_{BSS})) = P_1 \oplus P_2$
  - 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 $C = RC4(IV, K_{BSS}) \oplus (M, CRC(M))$
- The attacker E can construct a new ciphertext C' that will decrypt to a message M' with a valid checksum CRC(M'):
  - E chooses an arbitrary message Δ of the same length
  - $C' = C \oplus (\Delta, CRC(\Delta)) = RC4(IV, K_{BSS}) \oplus (M, CRC(M)) \oplus (\Delta, CRC(\Delta))$
    $= RC4(IV, K_{BSS}) \oplus (M \oplus \Delta, CRC(M) \oplus CRC(\Delta))$
    $= RC4(IV, K_{BSS}) \oplus (M \oplus \Delta, CRC(M \oplus \Delta))$
    $= RC4(IV, K_{BSS}) \oplus (M', CRC(M'))$
  - Note, that E does not know M' as it does not know M
  - Nevertheless, a "1" at position *n* in Δ results in a flipped bit at position *n* in M', so E can make controlled changes to M

  ⇒ 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 another 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 a related-key attack, exploiting WEP's usage of RC4:
    - RC4 is vulnerable to deducing bits of a key if:
      - many messages are encrypted with key stream 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 key stream 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] and was later refined to work even faster [TWP07]
  - 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

- Original 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 key stream 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, ...

Interlude: Security in Public WLAN Hotspots

What security can you expect in a public WLAN hotspot?

- For most hotspots: Unfortunately almost none!
- If you do not have to configure any security parameters besides typing in a username and password in a web page, expect the following:
  - The hotspot operator checks your authenticity at logon time (often protected with SSL to protect against eavesdropping on your password)
  - Only authenticated clients will receive service as packet filtering is deployed to only allow accessing the logon page until successful authentication
  - Once logon authentication has been checked: no further security measures
  - No protection for your user data:
    - Everything can be intercepted and manipulated
    - However, you can deploy your own measures, e.g. VPN or SSL, but configuration is often tedious or not even supported by communication partner and performance is affected because of additional (per-packet-) overhead
  - Plus: your session can be stolen by using your MAC & IP addresses!
- Consequence: better WLAN security is urgently required

Fixing WLAN Security: IEEE 802.11i, WPA & WPA2

- Scope: Defining the interaction between 802.1X and 802.11 standards
- TGi 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 (marketing name WPA)
    - AES encryption - robust data privacy for long term (marketing name WPA2)
  - Security association management:
    - Enterprise mode – based on 802.1X
    - Personal mode – based on pre-shared keys

(most material on 802.11i is taken from [WM02a])

WPA Key Management (I)

- In contrast to original 802.11: pair-wise keys between STA and BS, additional group keys for multi- and broadcast packets, as well as station-to-station link (STSL) keys
- The first secret: the 256 bit *Pairwise Master Key (PMK)*
  - Enterprise mode: Uses 802.1X authentication and installs a new key known to BS and client, e.g., by EAP-TTLS
  - Personal mode: Uses pre-shared key (*PSK)* known to BS and many STAs
    - Explicitly given by 64 random hex characters or implicitly by password
    - If password: PMK = PBKDF2(password, SSID, 4096, 256)
    - Where PBKDF2 is the Password-Based Key Derivation Function 2 from [RFC2898] with a *salt* SSID and 256 bit output length
    - Implies 2 * 4096 calculations of HMAC-SHA1 to slow down brute-force

- PMK is trust anchor to run authentication by EAPOL (EAP over LAN) handshake, but will never be used directly…

WPA Key Management (II)

- For actual cryptographic protocols a short-term 512 bit *Pairwise Transient Key (PTK)* is generated by
  - *PTK = PRF(PMK, "Pairwise key expansion", min($Addr_{BS}$, $Addr_{STA}$) ||* *max($Addr_{BS}$, $Addr_{STA}$) || min(r_{BS}, r_{STA}) || max(r_{BS}, r_{STA}))*
  - Where PRF(K, A, B) is the concatenated output of *HMAC-SHA1(K, A* || *'0'* || *B* || *i)* over a running index *i*
- The PTK is split into:
  - EAPOL Key Confirmation Key (KCK, first 128 bits),
    - Used to integrity protect EAPOL messages
    - By HMAC-MD5 (deprecated), HMAC-SHA1-128, AES-128-CMAC
  - EAPOL Key Encryption Key (KEK, second 128 bits),
    - Used to encrypt new keys in EAPOL messages
    - By RC4 (deprecated), AES in Key Wrap Mode [RFC3394]
  - A Temporal Key (TK) to protect data traffic (starting from bit 256)!

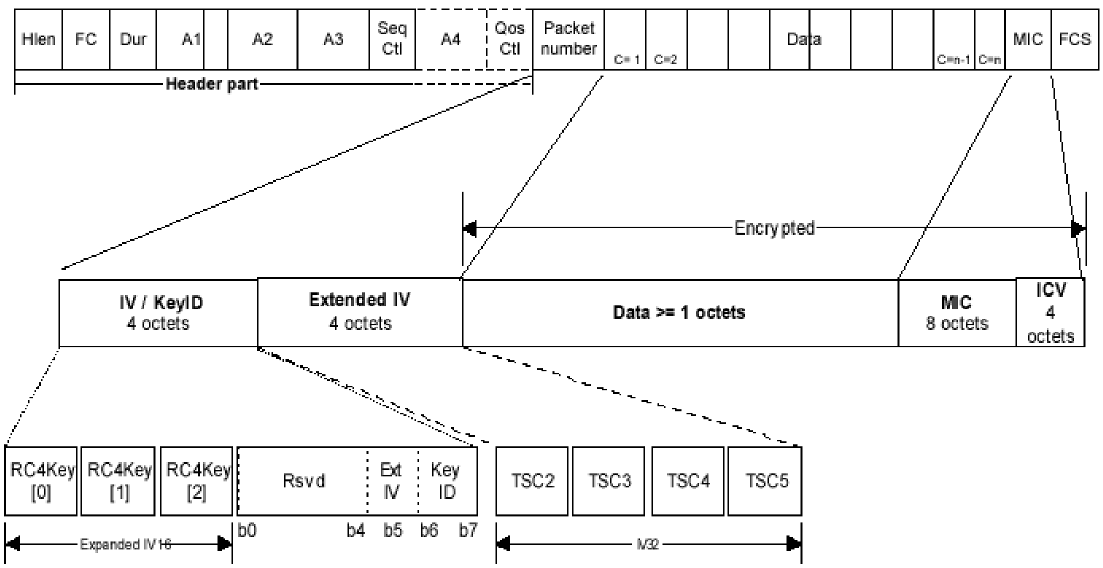
WPA Key Management (III)

- Initial dialog with BS:
  - EAPOL (EAP over LAN) 4-way handshake is used to
    - Verify mutual knowledge of PMK
    - Initiated by BS to install keys (group and new pairwise)
  - Simplified handshake works as follows:
    1. BS → STA: (1, r_{BS}, PMKID, install new PTK)
    2. STA → BS: (2, $r_{STA,}$ MAC_{KCK})
    3. BS → STA: (3, r_{BS}, $MAC_{KCK,}$ $\{TK\}_{KEK}$)
    4. STA → BS: (4, $r_{STA,}$ MAC_{KCK})
    - Where PMKID identifies the PMK: Upper 128 bit of HMAC-SHA-256(PMK, "PMK Name" || $Addr_{BS}$ || $Addr_{STA}$)

An Intermediate Solution: Temporal Key Integrity Protocol

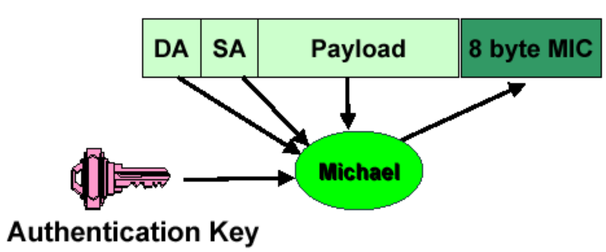
- Design Goals:
  - Quick fix to the existing WEP problem, runs WEP as a sub- component
  - Can be implemented in software, reuses existing WEP hardware
  - Requirements on existing AP hardware:
    - 33 or 25 MHz ARM7 or i486 already running at 90% CPU utilization before TKIP
    - Intended to be a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



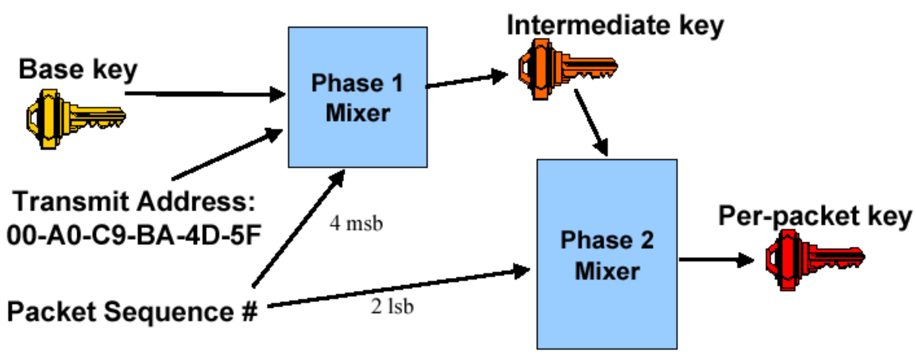
TKIP Design: Message Integrity Code Function Michael

- 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

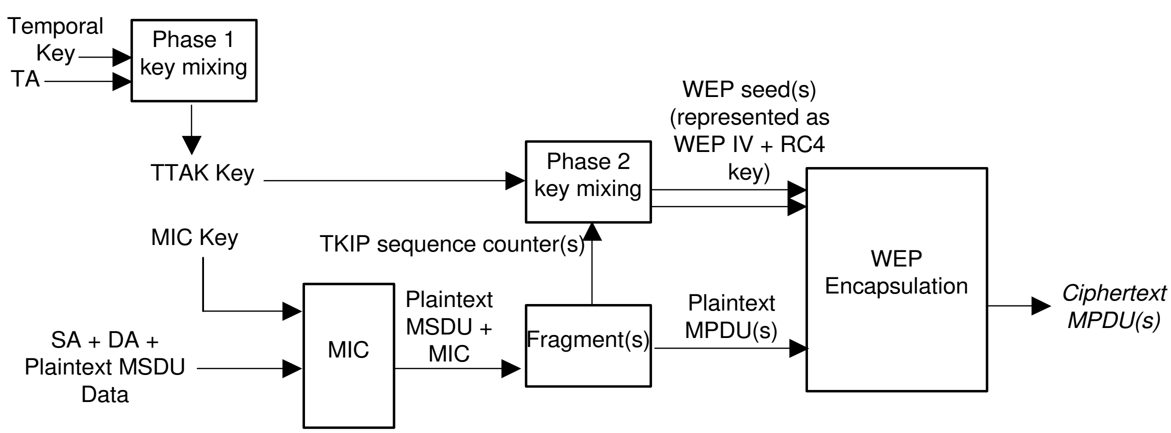


TKIP Design: Replay Protection and RC4 Key Scheduling

- 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

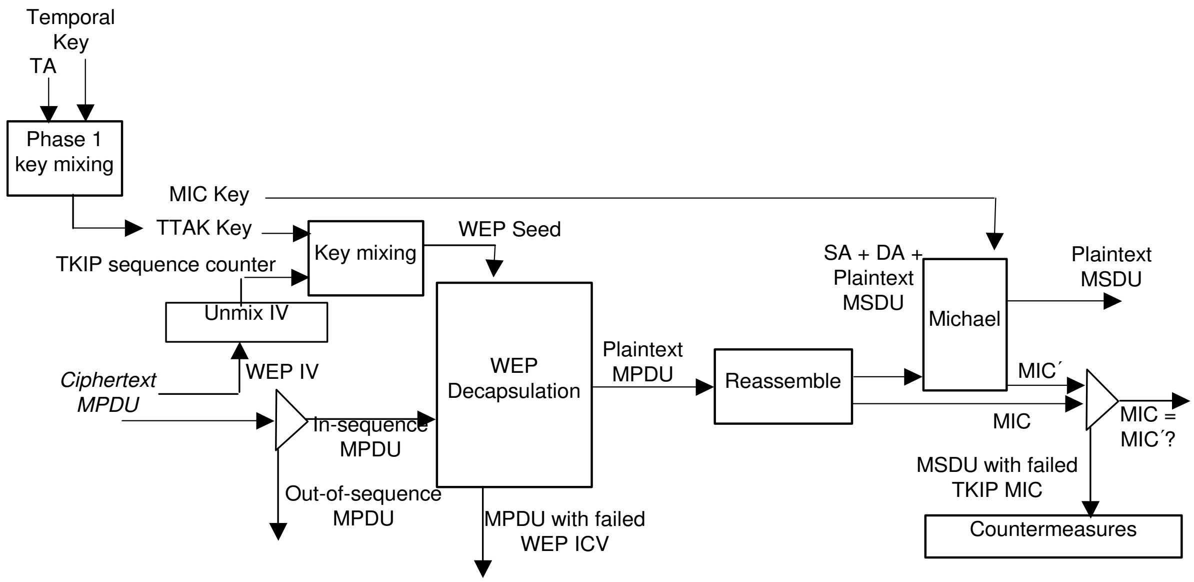


TKIP Processing at the Sender



(source: IEEE 802.11 Tgi draft)

TKIP Processing at the Receiver

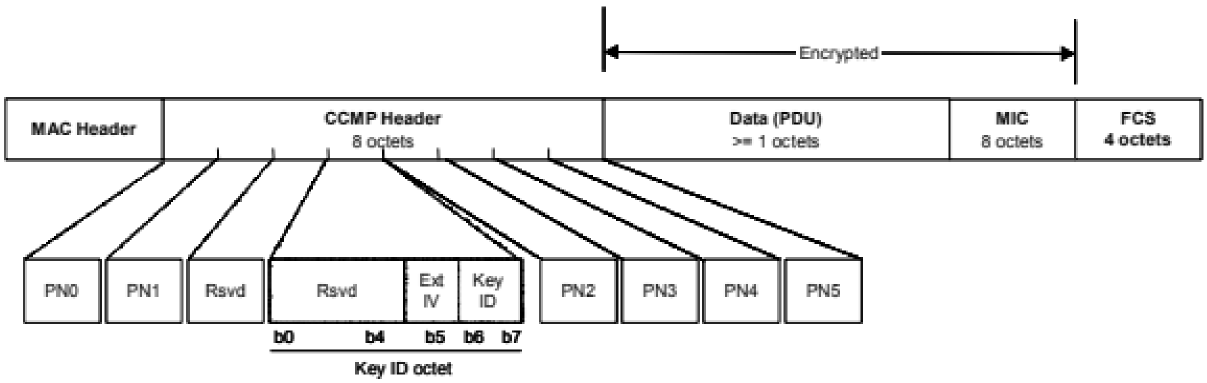


(source: IEEE 802.11 Tgi draft)

The Long Term Solution: AES based WLAN Protection

- Counter mode with CBC-MAC (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 in theory not PCs (however, this will increase CPU load and energy consumption), practically due to missing drivers for both

AES-CCMP: Frame format



Comparison of WEP, TKIP, and CCMP

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

→ Currently TKIP is deprecated, AES is recommended

Additional References

[BGW01a] N. Borisov, I. Goldberg, D. Wagner. *Intercepting Mobile Communications: The Insecurity of 802.11.* 7th ACM SIGMOBILE Annual International Conference on Mobile Computing and Networking (MOBICOM), Rome, Italy, July 2001.

[FMS01a] S. Fluhrer, I. Mantin, A. Shamir. *Weaknesses in the Key Scheduling Algorithm of RC4.* Eighth Annual Workshop on Selected Areas in Cryptography, August 2001.

[IEEE12] IEEE. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE Std 802.11-2012, The Institute of Electrical and Electronics Engineers (IEEE), 2012.

[Riv01a] R. Rivest. *RSA Security Response to Weaknesses in Key Scheduling Algorithm of RC4.* http://www.rsa.com/rsalabs/technotes/wep.html, 2001.

[SIR01a] A. Stubblefield, J. Ioannidis, A. D. Rubin. *Using the Fluhrer, Mantin, and Shamir Attack to Break WEP.* AT&T Labs Technical Report TD-4ZCPZZ, August 2001.

[TWP07] E. Tews, R. P. Weinmann, A. Pyshkin. *Breaking 104 bit WEP in less than 60 seconds.* Information Security Applications, 188-202, 2007.

[WM02a] N. C. Winget, T. Moore, D. Stanley, J. Walker. *IEEE 802.11i Overview.* NIST 802.11 Wireless LAN Security Workshop, Falls Church, Virginia, December 4-5, 2002.

Additional References

[RFC2898] B. Kaliski. *PKCS #5: Password-Based Cryptography Specification Version 2.0.* IETF Request for Comments 2898, 2000.

[RFC3394] J. Schaad, R. Housley. *Advanced Encryption Standard (AES) Key Wrap Algorithm.* IETF Request for Comments 3394, 2002.