Csci388 Wireless and Mobile Security – Temporal Key Integrity Protocol

Xiuzhen Cheng
cheng@gwu.edu

Introduction

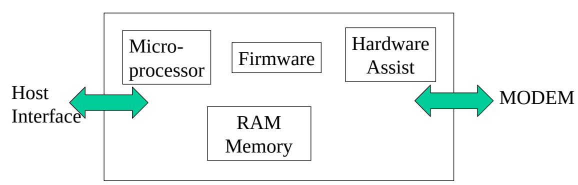
- TKIP has been adopted as part of WPA certification
- A part of RSN in 802.11i
- TKIP is used with existing Wi-Fi equipment
- Purpose:
  - To allow WEP system to be upgraded to be secure – backward compatibility
  - To address all the known attacks and deficiencies in WEP
  - It significantly improves WEP, and yet is able to operate on the same type of hardware (support RC4, Not AES) and can even be applied to many older Wi-Fi systems through firmware upgrades
- The design of TKIP has a severe restriction in hardware, it should be secure and available as an upgrade to WEP system
- CCMP is designed from scratch

Weaknesses of WEP – Revisited

1. IV is too short and not protected from reuse
2. The per packet key is constructed from the IV, making it susceptible to weak key attacks
3. No effective detection of message tampering (message integrity)
4. Master key is used directly and no built-in provision to update the keys
5. There is no protection against message replay

Inside the MAC Chip in Wi-fi Cards

- WEP depends on the Hardware Assist to achieve high data rate!
- The Hardware Assist support RC4 only, not AES-CCMP!



Changes from WEP to TKIP

- Message Integrity: add a message integrity protocol to prevent tampering that can be implemented in software (3)
- IV selection and reuse: Change the rules of IV selection and IV is reused as a replay counter (1, 3, 5)
- Per-Packet Key Mixing: change the encryption key for every frame (1,2,4)
- IV Size: Increase the size of IV to avoid ever reusing the same IV (1,4)
- Key Management: Add a mechanism to distribute and change the broadcast keys -- the key hierarchies (4)

Message Integrity

- The ICV based on CRC still computed, but not used for integrity check
- TKIP: compute MIC (Message Integrity Code) transmitted with the message
- Existing well-known MIC computation methods are not applicable to TKIP
  - They require either multiplication or new cryptographic algorithms
  - The microprocessor inside the MAC chip of most Wi-Fi cards is not very powerful
  - It does not have any sort of fast multiplication
  - A 32-bit multiply may take 50 microseconds to compute
  - This reduces the data throughput of 802.11b from 11Mbps to 1Mbps
  - Move the MIC computation up to the software driver level does not work since some old AP does not have the high-power processor

Message Integrity

- **MIC computation using Michael is adopted**
  - by cryptographer Niels Ferguson, designed specifically for TKIP
  - No multiplication, just shift and add operations
  - Fit in the existing AP – the purpose of TKIP
  - Check word is short (equivalent to 20-bit of security), suffering from the brute force attacks
  - Countermeasures are introduced against the brute force attacks
  - Michael operates on MSDUs, not MPDUs.
    - Done at the upper layer (at the device driver)
    - Reduce overhead – don't do for each MPDU
  - Michael uses a different key than encryption
- **A simple Countermeasure**
  - Use a reliable method for attack detection
  - When attacks are detected, shut down the communication for 1 minute
    - Regenerate keys
    - Limit the attacker to one try per minute for the entire network

Message Integrity – Michael

- **A 20-bit of security means once in a million times the attacker can win (without being detected after message modification)**
- **Shut down the communication to the attacked station for 1 minute limits the attacker one try per minute**
  - Disable the keys for a link as soon as the attack is detected
  - The new keys are generated until the 60-second period has expired
- **MIC failure can be detected at the mobile device and at the access point**

Computation of MIC

- **Only substitutions, rotations, and XOR operations are involved**
- **Used 64-bit key to generate 64-bit MIC**
  - The 128-bit temporal MIC key is divided into two parts, one used by the supplicant (user), and one by the authenticator (AP)
  - Michael takes only 64-bit key
- **Michael requires the length of the packet in bytes be a power of 4, and the last 4 bytes must contain all 0s**
  - The first padding byte must be 0x5a
- **Michael algorihtm**
  - Based on 32-bit words

IV Selection and Use

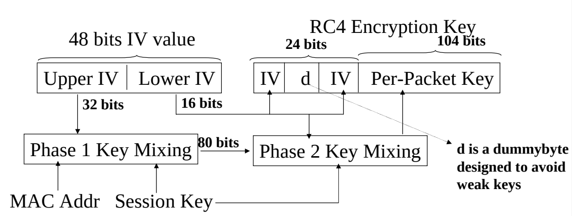
- **Weakness of IV in WEP**
  - Too short, only 24 bits, reuse is common
  - IV is not bounded to the station – same IV can be used with the same key on multiple wireless devices
  - IV is prepended to the key, making it susceptible to weak key attacks
- **IV in TKIP**
  - Size is increased from 24 to 48 bits
  - IV has a second role as a sequence counter to avoid replay attacks
  - IV is constructed in a way to avoid certain "weak keys"
  - IV is not protected by MIC!
    - Replay old frames with new IV value, causing Denial of service attack!

IV Size Increase

- **Insert 32 bits in between the existing WEP IV and the start of the encrypted message**
  - Contentious, since not all vendors can upgrade their legacy systems to support this requirement
- **One byte is thrown away to avoid weak keys**
- **IV value rollover**
  - For 24 bits, an IV will be reused after 16777216 packets if IV value is incremented by 1 each time
  - For a device sending 10000 packets per second
    - 24-bit IV takes half an hour to rollover
    - 48-bit IV takes 900 years!

48-bit IV in TKIP

- **WEP per packet key is 24+104 bits, how to handle the 48+104=152 bits key with existing WEP?**
  - Per-packet key mixing
  - The value of the key used for RC4 encryption is different for every IV value
  - The structure of the RC4 key is a 24-bit "old IV" field and a 104-bit secret key field



Per-Packet Key Mixing

- **Motivation**
  - Against RC4 weak key attacks. Dropping the first 256 bytes of the key stream is not supported by the hardware
  - Incorporate the extra bits in the extended IV
- **Idea**
  - Combine the session key, the IV, and the source MAC address with a hash function to produce a mixed key
  - Why include MAC address? – separate the key space; forgery protection; otherwise, IV collision if A and B sends to each other with the same IV and the same session key
  - Practicality: Hash operation is too expensive for low-power MAC processor – use two phases pre-compute per-packet key since IV values increased monotonically
  - Efficiency consideration – two phases

Per-Packet Key Mixing

- **Both phases utilize a partial S-table containing 512 word entries, only shift, add, and XOR are involved**
- **Phase I**
  - Input: the 128-bit session key, the upper 32-bit of the IV, and the MAC address
  - Output: 80 bits
  - Computed once for 2^{16} packets
- **Phase II**
  - Input: the output of Phase I, the lower 16-bit of the IV, and the 128-bit session key
  - Output: the 128 bit encryption key
  - Can be precomputed
- **RC4 per packet key:**
  - The first and the third byte come from the lower 16-bit of the IV
  - The second byte is a repeat of the first byte, except that bit 5 is forced to 1 and bit 4 is forced to 0. This design can prevent the generation of the major class of weak keys
  - Nobody knows all weak keys!

IV as a Sequence Counter – the TSC

- **TSC refers to TKIP Sequence Counter**
- **TSC is used to prevent replay attacks**
  - With the same session key (temporary key), IV monotonically increases from 0
- **How?**
  - Throw out any message that has a TSC less than or equal to the last message? – how about retransmitted ones?
  - Burst-ACK: sending 16 packets in quick succession and waiting for the ACK of all packets within one ACK
    - Not adopted by 802.11 now but is likely to be adopted
  - Replay Window: keep the last 16 TSC values received
  - Packet rejection rule
    - ACCEPT: TSC is larger than the largest seen so far
    - REJECT: TSC is smaller than the largest 16 in the window
    - WINDOW: o.w. put in the window and adjust the window

Countering the Weak Key Attacks

- **Weak Key revisited**
  - The first few bytes of RC4 key stream are not random if a weak key is used
  - By exploring weak keys, it is possible to guess the encryption key
  - Rivest suggested a simple solution: throwing away the first 256 bytes of the random stream
  - However, RC4 is implemented in hardware in WEP, which does not support this solution
  - The prepending of IV values in WEP make the problem even worse – hard to avoid weak keys
- **Considerations in TKIP**
  - Try to avoid the weak keys
  - Try to further obscure the secret key

Countering the Weak Key Attacks

- **Change the secret encryption key for each packet**
  - Not just the IV, but the key!
- **Avoid using weak keys!**
  - Not practical, since no one knows all weak keys – can you prove a key is strong?
  - Best effort!
  - The current design (the dummy byte) can avoid a well-known class of weak keys

IV Summary

- **The length is increased to 48 bits**
- **IV is used as a sequence counter for replay attacks**
- **The last two bytes of the IV are used to form the WEP encryption key, with a dummy byte in between to counter the weak key attacks**