Securing Wireless LANs: Part 5

Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

References

- J. Kurose, K. Ross, "*Computer Networking: A Top Down Approach*", Sixth Edition, Pearson Education, 2013
- J. Edney, W.A. Arbaugh, "*Real 802.11 Security: Wi-Fi Protected Access and 802.11i*", Pearson Education, 2004.
- B.L. Menezes, R. Kumar, "*Cryptography, Network Security, and Cyber Laws*", CengageLearning India Pvt.Ltd., 2018

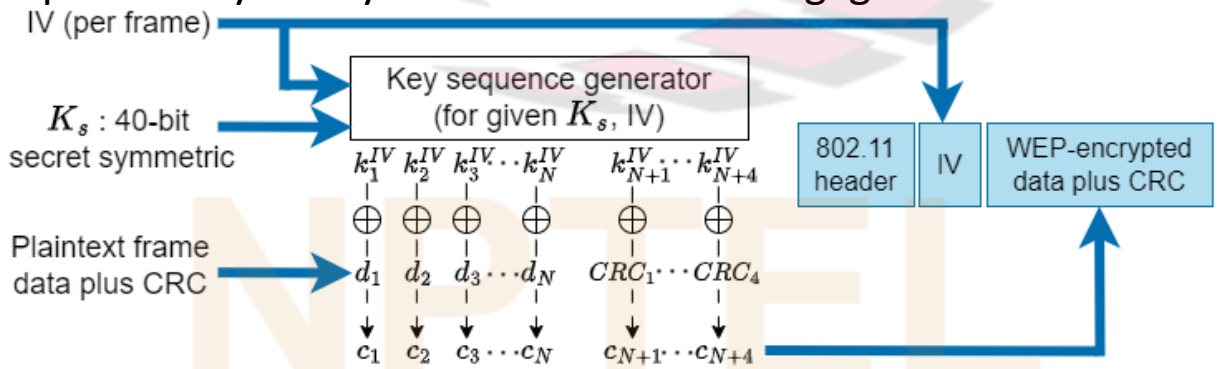
Confidentiality and Message Integrity in WPA and 802.11i

TKIP and CCMP

- Recall: RC4, as used in WEP, has several weaknesses
- This prompted the 802.11i standards committee to seek a replacement
- The new standard, AES, for symmetric key cryptography, was an obvious choice
- However, use of AES necessitates use of new hardware
  - when weaknesses were found in WEP, there were millions of installed Wi-Fi systems, which were rendered insecure
  - intermediate solution was needed that would allow them to be upgraded and become secure again
  - intermediate solution needed to run on already installed hardware
- Intermediate solution developed was:
  - firmware upgrade that retained existing 802.11 hardware (including RC4),
  - but with many changes in overall design that eliminated the vulnerabilities in WEP
- Intermediate solution known as *Temporal Key Integrity Protocol* (TKIP)
  - used for encryption and message integrity in WPA
- Final solution in 802.11i, which uses AES, is referred to as *Counter Mode with CBC-MAC Protocol (CCMP)*
  - used for encryption and message integrity in WPA2

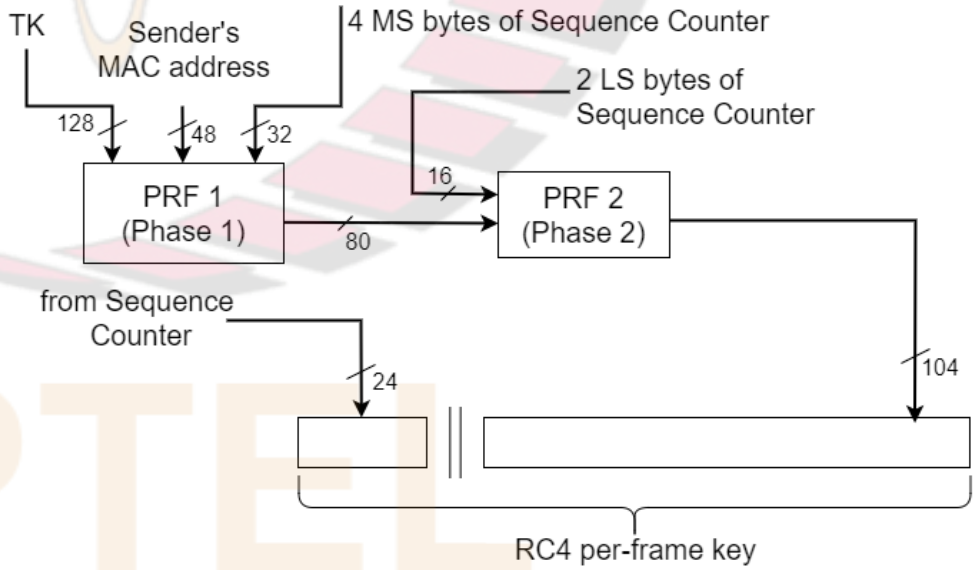
TKIP

- Recall: protocol used by WEP:
  - A 40-bit symmetric shared key, K_S, assumed to be known by both host and AP
  - A 24 −bit IV appended to K_S to get a 64-bit key that is used to encrypt a single frame; IV changes from frame to frame (*e.g.*, selected randomly)
  - The 64-bit key used to generate a stream of key values (1 byte each), k_1^{IV}, k_2^{IV}, k_3^{IV},… using RC4 stream cipher
- Drawback:
  - variable part of the WEP encryption key (the above 64-bit key) is too small (only 24 bits in length)
  - so per-frame keystream repeats frequently in a busy network
  - if keystreams of two frames are same, then intruder can get XOR of plaintext of the two frames by XORing the ciphertext; if some of the bits of plaintext known, then others can be deduced
- In contrast, encryption key in TKIP is 128 bits long
- More importantly, method used to generate it is much more sophisticated (see next slide)
  - designed such that there is a lot of randomness in most of the 128 bits of the key and hence the probability of keystream collisions is negligible



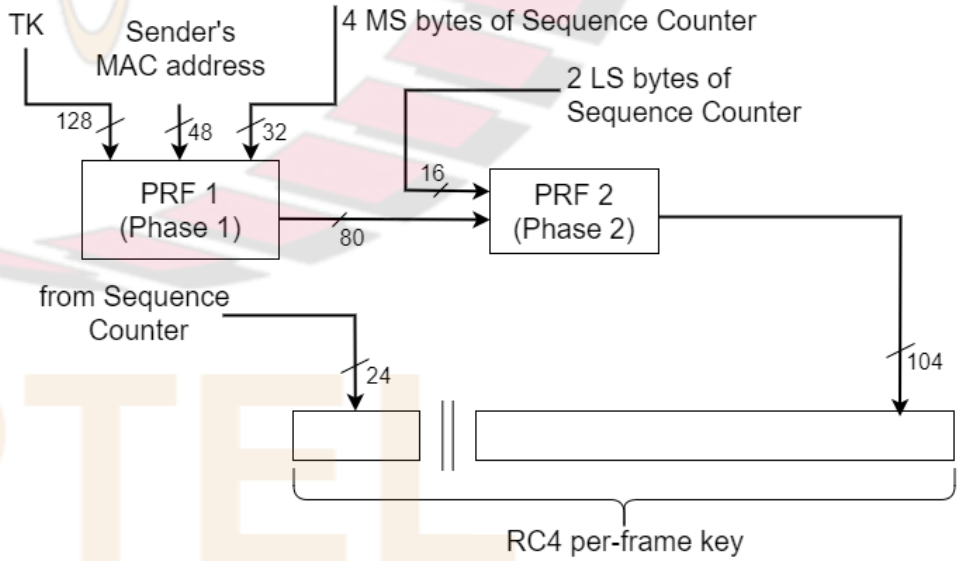
Generation of Encryption Key in TKIP

- Fig. shows process used to generate encryption key in TKIP
  - ❑ called "*two-phase key mixing*"
- Inputs to the process are the 128-bit Temporal Key (TK), sender's MAC address, and a 48-bit frame sequence counter
- Sequence counter is incremented for every frame sent; carried in header of each frame
- Two pseudo-random functions (PRF1 and PRF2) are used in the two phases
  - ❑ The least significant 16 bits of the sequence counter are inputs to PRF2
  - ❑ So output of PRF2 changes for every frame sent
  - ❑ The 32 most significant bits of the sequence counter are input to PRF1
  - ❑ This input changes after every $2^{16} = 65{,}536$ frames sent
  - ❑ Hence, PRF1 is executed very rarely and overall computation time is saved
- The randomizing properties of the key mixing function and the large size (128 bits) of the key space ensure that keystream collisions occur very rarely



Generation of Encryption Key in TKIP (contd.)

- Encryption in TKIP is done as in WEP, with the change that instead of the 64-bit encryption key (K_S+IV), the 128-bit output of the above two-phase key mixing function is used as the encryption key for RC4 keystream generation
- Recall:
  - it was shown that when WEP is used, an attacker can find the secret key (used for encryption) in a small amount of time after eavesdropping on the network
  - depending on the amount of network traffic, a successful key recovery may take as little as one minute
  - automated tools are available on the Internet that implement above attack
- It can be shown that the above *two-phase key mixing function used in TKIP eliminates the WEP key recovery attacks*
  - details omitted



Message Integrity in TKIP

- Recall:
  - ❑ method used to compute the 4-byte CRC in WEP such that it is possible to predict which bits in the CRC will change if we change a single bit in the data message
  - ❑ we discussed earlier that due to above property, intruder can break message integrity of WEP
- Above property arises due to the fact that the *CRC in WEP is a linear function,* i.e., $CRC(m_1 \oplus m_2) = CRC(m_1) \oplus CRC(m_2)$
- 64-bit message integrity check in TKIP, called MIC or Michael, is a significant improvement over the CRC in WEP
- Unlike the CRC, MIC is non-linear, i.e.:
  - ❑ $MIC(m_1 \oplus m_2) \neq MIC(m_1) \oplus MIC(m_2)$
- MIC is computed as a function of the data in the frame and some fields in the header such as the source and destination addresses
  - ❑ it also uses as input a key derived from the PTK, which was computed during the four-way handshake
- Due to limitations of WEP hardware, MIC uses only simple logical functions, shift and add
  - ❑ no multiplications are used since they are computationally expensive
- Hence, MIC is not as secure as a keyed cryptographic hash
- On the other hand, it is much stronger than the CRC checksum used in WEP

Replay Attack Prevention in TKIP

- Recall:
  - WEP provides no protection against replay attack
- Replay attack prevention is provided in TKIP as follows
- A 48-bit frame sequence counter is carried in header of each frame
  - starts from 0 and increments by 1 for every packet sent
  - unlikely to wrap around since it is 48-bits long; if it gets close to wrapping around, then client renegotiates a new PTK
- Extracted by receiver and used to compute the RC4 key for decryption
- Sender and receiver keep track of the sequence number of the last frame sent/ received
  - receiver accepts a frame only if its sequence number is greater than that of the previous correctly received frame
- Hence, if intruder replays an old frame, then it is rejected by receiver
- Suppose an intruder creates a new frame with a higher sequence number than sequence number of last frame sent, and sends it to receiver
- Will receiver accept it?
  - no, since RC4 key for generation of keystream is a function of sequence number and temporal key (which is unknown to intruder)
  - so after receiver computes RC4 key and decrypts frame, MIC verification will fail
- Thus, TKIP defends against replay attack

Security of TKIP

- As discussed above, TKIP is much more secure than WEP
- Incorporates mechanisms to defend against several attacks to which WEP is vulnerable, e.g.:
  - key recovery attacks
  - attacks on message integrity of packets
  - replay attacks
- However, it has been shown that TKIP is vulnerable to some new attacks
  - details omitted
- Hence, TKIP is no longer considered secure, and was deprecated in the 2012 revision of the 802.11 standard