Aircrack

by Christopher Primerano and Yuliya Cherenkova

WEP

(Wired Equivalent Privacy)

- Two methods of authentication: Open System authentication and Shared Key authentication

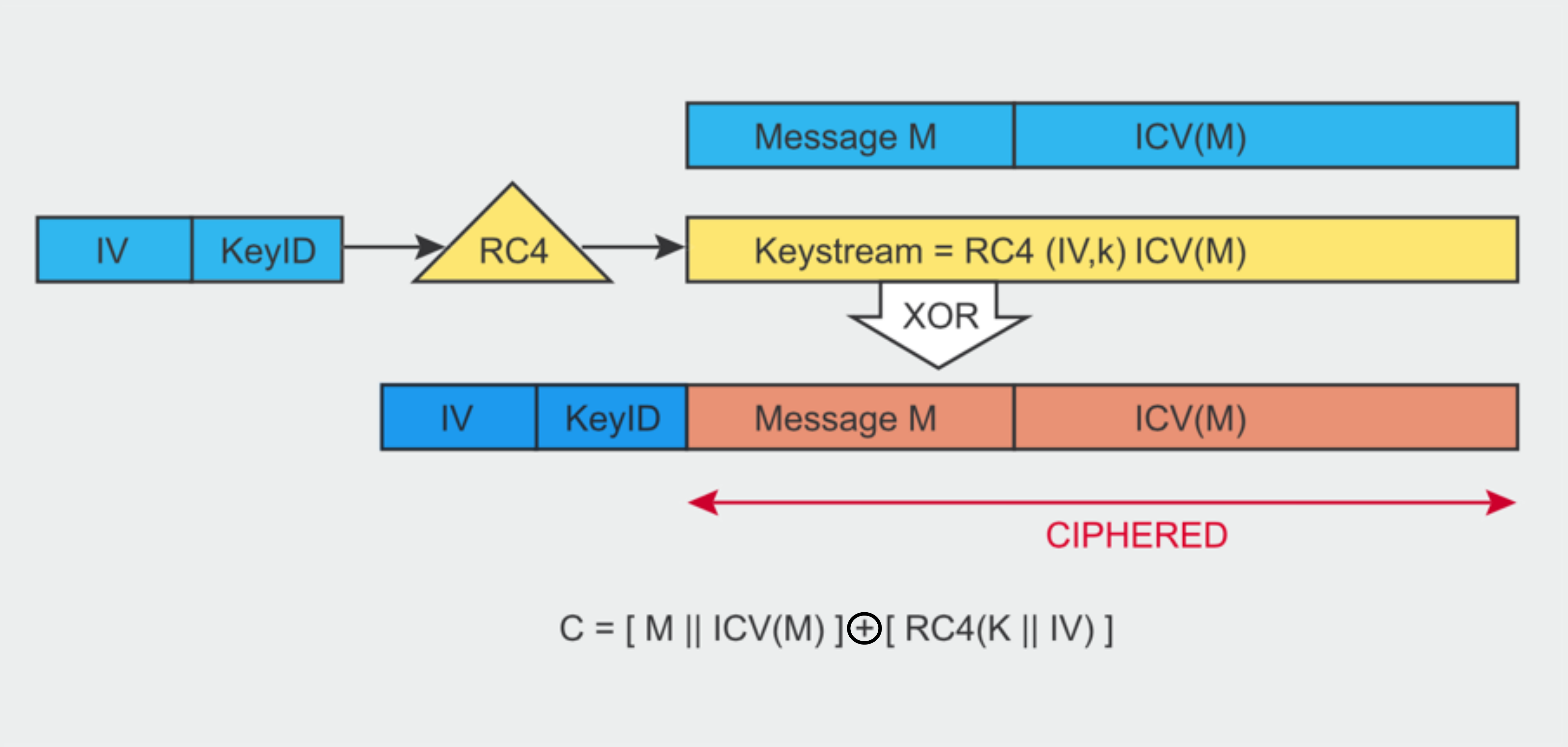
- RC4 encryption algorithm (a stream cipher)

- Pre-shared keys: to avoid using same key with a stream cipher, WEP concatenates a 24-bit initialization vector (IV) with the key.

- To ensure that a packet has not been modified in transit, it uses an Integrity Check (ICV) field in the packet, containing CRC-32 checksum

WEP

(Wired Equivalent Privacy)



$$C = [\, M \,||\, ICV(M) \,] \oplus [\, RC4(K \,||\, IV) \,]$$

WEP encryption protocol

WEP

Security Flaws

- Shared Key authentication is one-way: a client cannot verify AP
- Open System: any client can connect to the AP
- CRC-32 is not cryptographically secure.
- No built-in method of updating keys

WEP

Security Flaws

- The 802.11 standard does not specify how the IVs are set or changed. IV reuse is allowed

- Birthday paradox: 24 bits in IV

- The IV is a part of the RC4 encryption key and it is sent in plaintext

- RC4 algorithm weaknesses within the WEP protocol due to key construction: certain IV values yield weak keystreams

WEP

Passive Attack to Decrypt Traffic

- Observe traffic until IV collision occurs

- XOR two packets that use the same IV to obtain the XOR of the two plaintext messages

- IP traffic is often very predictable and includes a lot of redundancy, which is helpful for statistical analysis

- If an attacker can send traffic from a host on the web to the host on the target network, it gets very easy

WEP

Active Attack to Inject Traffic

- An attacker can construct correct encrypted packets knowing exact plaintext of one encrypted message
- Construct a new message, calculate CRC-32, and perform bit flips on the original encrypted message
- Based on: C(X) xor X xor Y = C(Y)

WEP

Current State

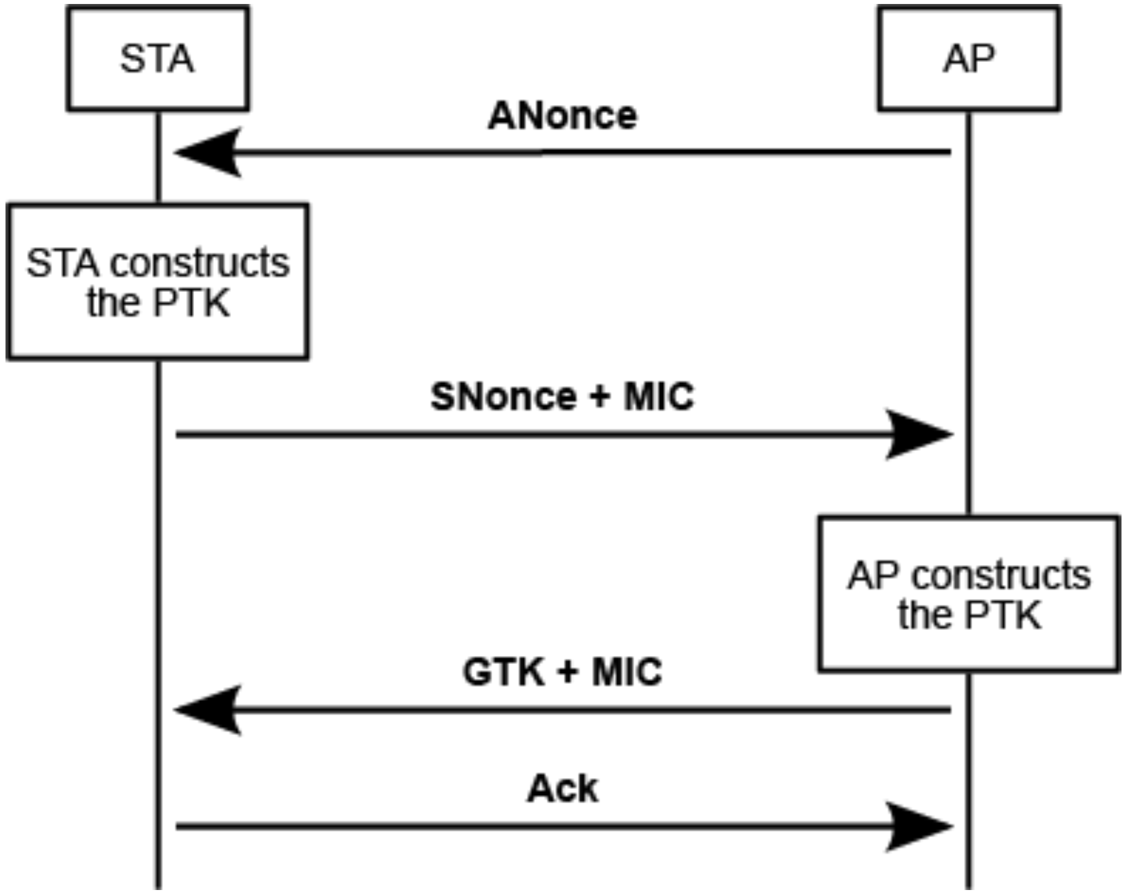
- No longer included as a possible security option on new routers

- Is not a valid security option when setting up a wireless N or AC network

WPA / WPA 2

WPA Handshake

- ANonce, SNonce - Randomly generated
- MIC (Message Integrity Code) - Computed from message
- All plaintext



WPA Keys

- PSK (Pre-Shared Key) - This key is generated from the passphrase and SSID when a network is created

- PMK (Pairwise Master Key) - In a PSK environment this is equal to the PSK

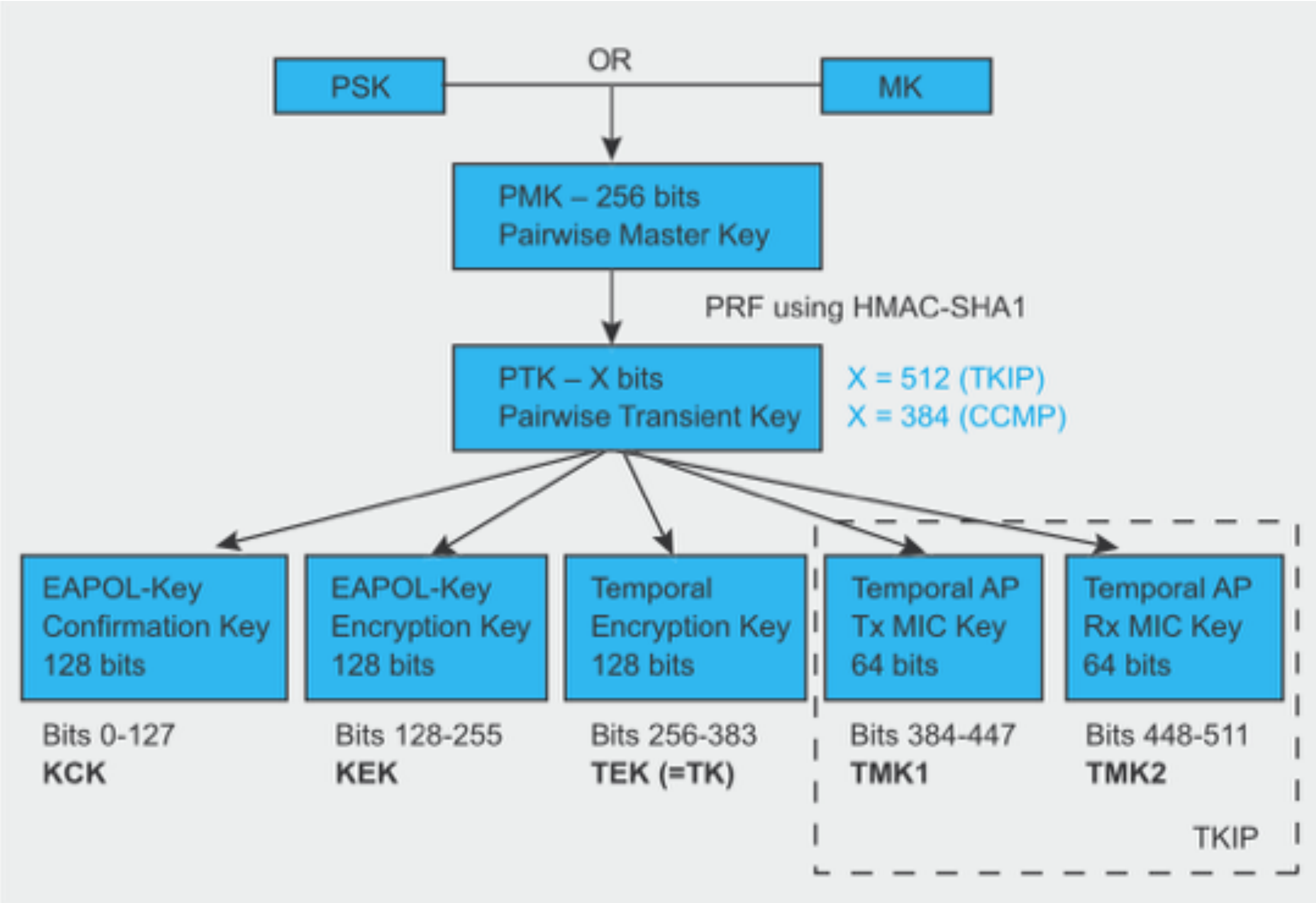
- PTK (Pairwise Transient Key) - This is generated from the PMK, the AP and Station MAC's and the AP and Station Nonce's running through a Pseudo Random Function

- PSK = PMK = PBKDF2(HMAC–SHA1, passphrase, SSID, 4096, 256)

- PTK = PRF(PMK, Min(AP_Mac, STA_Mac) || Max(AP_Mac, STA_ Mac) || Min(ANonce, SNonce) || Max(ANonce, SNonce))

WPA Keys

- 5 Keys from PTK
- KCK (Key Confirmation Key) is used to validate a MIC
- KCK is used for cracking WPA



WPA Cracking

- Need to capture 4-Way handshake
- From handshake [AS]Nonce, MAC’s and MIC are extracted
- Generate PMK from passphrase
- Compute PTK using [AS]Nonce, MAC’s
- Extract KCK and try to validate captured MIC
- If MIC validates then PMK is correct and so is passphrase
- If MIC does not validate, try new PMK

Aircrack

- A suite of tools focused on WiFi cracking
- airmon-ng - Adapter Mode Manager
- airodump-ng - Packet Capture
- aireplay-ng - Packet Injection
- aircrack-ng - Key Cracking
- etc

Demo

Demo

- airmon-ng start <interface>
- airodump-ng --output-format pcap -w <filename> --bssid <bssid> --channel <channel> mon0
- aireplay-ng -0 1 -a <bssid> -c <client mac> mon0
- aircrack-ng <filename> -w <dictionary>
- hashcat -m 2500 -a3 --pw-min=8 <filename> <pattern>