WiFi Security:
WEP, WPA, and WPA2

- security requirements in wireless networks
- WiFi primer
- WEP and its flaws
- 802.11i
- WPA and WPA2 (RSN)

Why security is more of a concern in wireless?

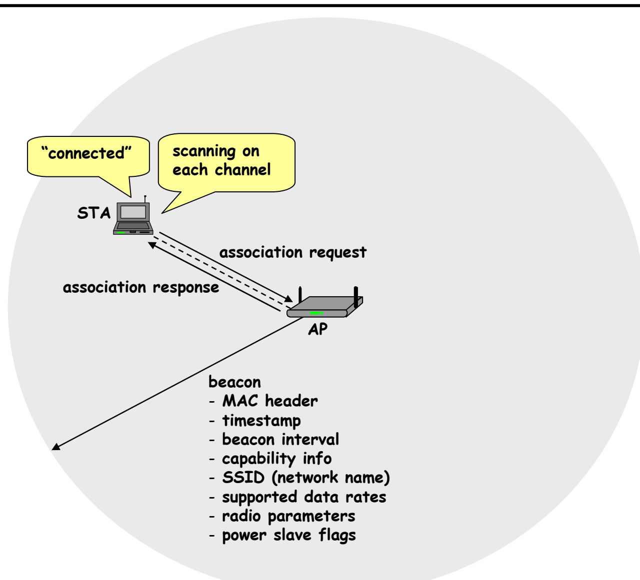
- no inherent physical protection
  - physical connections between devices are replaced by logical associations
  - sending and receiving messages do not need physical access to the network infrastructure (cables, hubs, routers, etc.)
- broadcast communications
  - wireless usually means radio, which has a broadcast nature
  - transmissions can be overheard by anyone in range
  - anyone can generate transmissions,
    - which will be received by other devices in range
    - which will interfere with other nearby transmissions and may prevent their correct reception (jamming)

- ➢ eavesdropping is easy
- ➢ injecting bogus messages into the network is easy
- ➢ replaying previously recorded messages is easy
- ➢ illegitimate access to the network and its services is easy
- ➢ denial of service is easily achieved by jamming

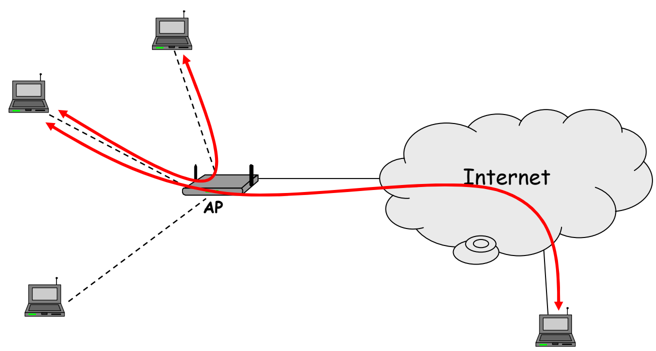
Wireless communication security requirements

- confidentiality
  - messages sent over wireless links must be encrypted
- authenticity
  - origin of messages received over wireless links must be verified
- replay detection
  - freshness of messages received over wireless links must be checked
- integrity
  - modifying messages on-the-fly (during radio transmission) is not so easy, but possible …
  - integrity of messages received over wireless links must be verified
- access control
  - access to the network services should be provided only to legitimate entities
  - access control should be permanent
    - it is not enough to check the legitimacy of an entity only when it joins the network and its logical associations are established, because logical associations can be hijacked
- protection against jamming

Introduction to WiFi



Introduction to WiFi



WEP – Wired Equivalent Privacy

- part of the IEEE 802.11 specification
- goal
  - make the WiFi network *at least as secure as a wired LAN* (that has no particular protection mechanisms)
  - WEP has never intended to achieve strong security
  - (at the end, it hasn't achieved even weak security)
- services
  - access control to the network
  - message confidentiality
  - message integrity

WEP – Access control

- before association, the STA needs to authenticate itself to the AP
- authentication is based on a simple challenge-response protocol:

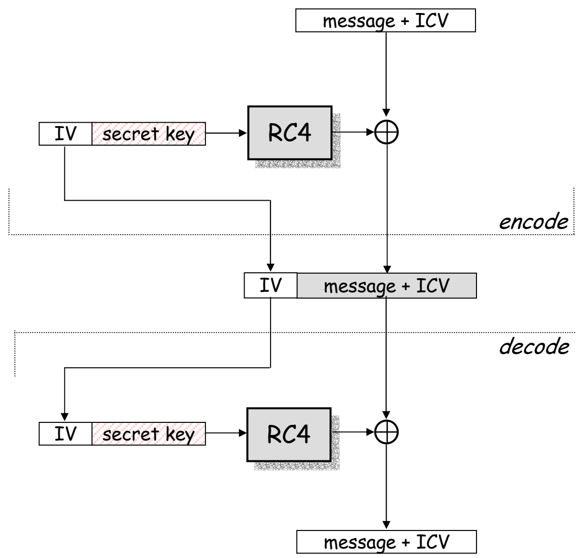
  STA → AP: authenticate request
  AP → STA: authenticate challenge (r)   // r is 128 bits long
  STA → AP: authenticate response ($e_K(r)$)
  AP → STA: authenticate success/failure

- once authenticated, the STA can send an association request, and the AP will respond with an association response
- if authentication fails, no association is possible

WEP – Message confidentiality and integrity

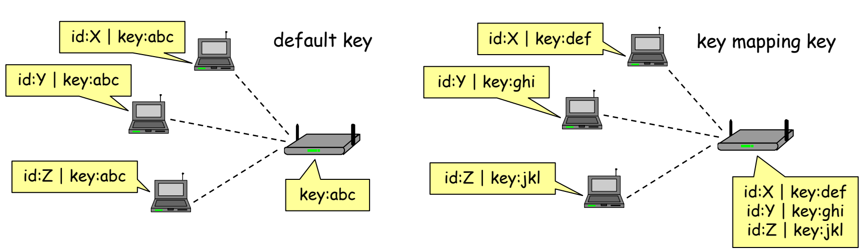
- WEP encryption is based on the RC4 stream cipher
  - operation:
    - for each message to be sent:
      - RC4 is initialized with the shared secret (between STA and AP)
      - RC4 produces a pseudo-random byte sequence (key stream)
      - this pseudo-random byte sequence is XORed to the message
    - reception is analogous
  - it is essential that each message is encrypted with a different key stream
    - the RC4 generator is initialized with the shared secret and an IV (initial value) together
      - shared secret is the same for each message
      - 24-bit IV changes for every message
- WEP integrity protection is based on an encrypted CRC value
  - operation:
    - ICV (integrity check value) is computed and appended to the message
    - the message and the ICV are encrypted together

WEP – Message confidentiality and integrity



WEP – Keys

- two kinds of keys are allowed by the standard
  - default key (also called shared key, group key, multicast key, broadcast key, key)
  - key mapping keys (also called individual key, per-station key, unique key)

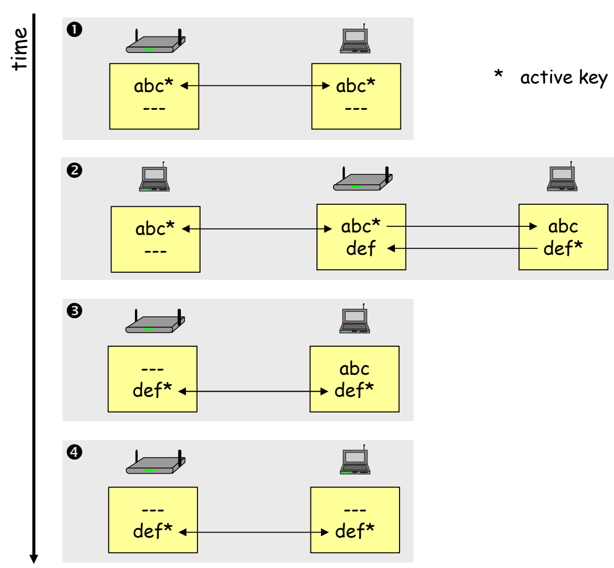


- in practice, often only default keys are supported
  - the default key is manually installed in every STA and the AP
  - each STA uses the same shared secret key → in principle, STAs can decrypt each other's messages

WEP – Management of default keys

- the default key is a group key, and group keys need to be changed when a member leaves the group
  - e.g., when someone leaves the company and shouldn't have access to the network anymore
- it is practically impossible to change the default key in every device simultaneously
- hence, WEP supports multiple default keys to help the smooth change of keys
  - one of the keys is called the active key
  - the active key is used to encrypt messages
  - any key can be used to decrypt messages
  - the message header contains a key ID that allows the receiver to find out which key should be used to decrypt the message

WEP – The key change process



WEP flaws – Authentication and access control

- authentication is one-way only
  - AP is not authenticated to STA
  - STA may associate to a rogue AP
- the same shared secret key is used for authentication and encryption
  - weaknesses in any of the two protocol can be used to break the key
  - different keys for different functions are desirable
- no session key is established during authentication
  - access control is not continuous
  - once a STA has authenticated and associated to the AP, an attacker send messages using the MAC address of STA
  - correctly encrypted messages cannot be produced by the attacker, but replay of STA messages is still possible
- STA can be impersonated
  - … next slide

WEP flaws – Authentication and access control

- recall that authentication is based on a challenge-response protocol:

  …
  AP → STA: r
  STA → AP: IV | r ⊕ K
  …

  where K is a 128 bit RC4 output on IV and the shared secret
- an attacker can compute r ⊕ (r ⊕ K) = K
- then it can use K to impersonate STA later:

  …
  AP → attacker: r'
  attacker → AP: IV | r' ⊕ K
  …

WEP flaws – Integrity and replay protection

- there's no replay protection at all
  - IV is not mandated to be incremented after each message
- attacker can manipulate messages despite the ICV mechanism and encryption
  - CRC is a linear function wrt to XOR:

$$CRC(X \oplus Y) = CRC(X) \oplus CRC(Y)$$

  - attacker observes $(M \mid CRC(M)) \oplus K$ where K is the RC4 output
  - for any ΔM, the attacker can compute $CRC(\Delta M)$
  - hence, the attacker can compute:

$$((M \mid CRC(M)) \oplus K) \oplus (\Delta M \mid CRC(\Delta M)) =$$
$$((M \oplus \Delta M) \mid (CRC(M) \oplus CRC(\Delta M))) \oplus K =$$
$$((M \oplus \Delta M) \mid CRC(M \oplus \Delta M)) \oplus K$$

WEP flaws – Confidentiality

- IV reuse
  - IV space is too small
    - IV size is only 24 bits → there are 16,777,216 possible IVs
    - after around 17 million messages, IVs are reused
    - a busy AP at 11 Mbps is capable for transmitting 700 packets per second → IV space is used up in around 7 hours
  - in many implementations IVs are initialized with 0 on startup
    - if several devices are switched on nearly at the same time, they all use the same sequence of IVs
    - if they all use the same default key (which is the common case), then IV collisions are readily available to an attacker
- weak RC4 keys
  - for some seed values (called weak keys), the beginning of the RC4 output is not really random
  - if a weak key is used, then the first few bytes of the output reveals a lot of information about the key → breaking the key is made easier
  - for this reason, crypto experts suggest to always throw away the first 256 bytes of the RC4 output, but WEP doesn't do that
  - due to the use of IVs, eventually a weak key will be used, and the attacker will know that, because the IV is sent in clear

→ WEP encryption can be broken by capturing a few million messages !!!

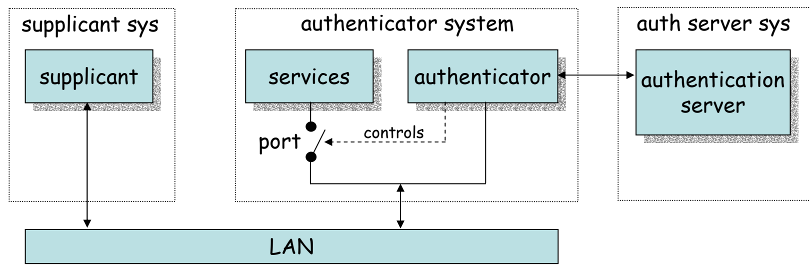
WEP – Lessons learnt

1. engineering security protocols is a **very** risky business
   - you may combine otherwise strong building blocks in a wrong way and obtain an insecure system at the end
     - example:
       - stream ciphers alone are OK
       - challenge-response protocols for entity authentication are OK
       - but they shouldn't be combined
     - example:
       - encrypting a message digest to obtain an ICV is a good principle
       - but it doesn't work if the message digest function is linear wrt to the encryption function
   - don't do it alone (unless you are a security expert)
     - functional properties can be tested, but security is a non-functional property → it is extremely difficult to tell if a system is secure or not
   - using an expert in the design phase pays out (fixing the system after deployment will be much more expensive)
     - experts will not guarantee that your system is 100% secure
     - but at least they know many pitfalls that you don't
     - they know the details of crypto algorithms better than you do
2. avoid the use of WEP (as much as possible)

Overview of 802.11i

- after the collapse of WEP, IEEE started to develop a new security architecture → 802.11i
- main novelties in 802.11i wrt to WEP
  - access control model is based on 802.1X
  - flexible authentication framework (based on EAP)
  - authentication can be based on strong protocols (e.g., TLS)
  - authentication process results in a shared session key (which prevents session hijacking)
  - different functions (encryption, integrity) use different keys derived from the session key using a one-way function
  - integrity protection is improved
  - encryption function is improved
- 802.11i defines the concept of RSN (Robust Security Network)
  - integrity protection and encryption is based on AES (in CCMP mode)
  - nice solution, but needs new hardware → cannot be adopted immediately
- 802.11i also defines an optional protocol called TKIP
  - integrity protection is based on Michael
  - encryption is based on RC4, but WEP's problems have been avoided
  - ugly solution, but runs on old hardware (after software upgrade)
- industrial names
  - TKIP → WPA (WiFi Protected Access)
  - RSN/AES-CCMP → WPA2

802.1X authentication model



- the supplicant requests access to the services (wants to connect to the network)
- the authenticator controls access to the services (controls the state of a port)
- the authentication server authorizes access to the services
  - the supplicant authenticates itself to the authentication server
  - if the authentication is successful, the authentication server instructs the authenticator to switch the port on
  - the authentication server informs the supplicant that access is allowed

Mapping the 802.1X model to WiFi

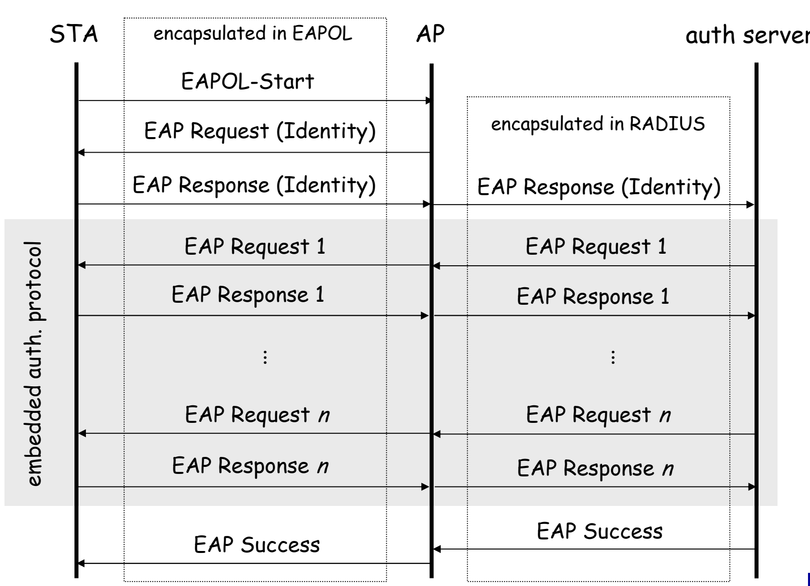
- supplicant → mobile device (STA)
- authenticator → access point (AP)
- authentication server → server application running on the AP or on a dedicated machine
- port → logical state implemented in software in the AP

- one more thing is added to the basic 802.1X model in 802.11i:
  - successful authentication results not only in switching the port on, but also in a session key between the mobile device and the authentication server
  - the session key is sent to the AP in a secure way
    - this assumes a shared key between the AP and the auth server
    - this key is usually set up manually

Protocols – EAP, EAPOL, and RADIUS

- EAP (Extensible Authentication Protocol) [RFC 3748]
  - carrier protocol designed to transport the messages of "real" authentication protocols (e.g., TLS)
  - very simple, four types of messages:
    - EAP request – carries messages from the supplicant to the authentication server
    - EAP response – carries messages from the authentication server to the supplicant
    - EAP success – signals successful authentication
    - EAP failure – signals authentication failure
  - authenticator doesn't understand what is inside the EAP messages, it recognizes only EAP success and failure
- EAPOL (EAP over LAN) [802.1X]
  - used to encapsulate EAP messages into LAN protocols (e.g., Ethernet)
  - EAPOL is used to carry EAP messages between the STA and the AP
- RADIUS (Remote Access Dial-In User Service) [RFC 2865-2869, RFC 2548]
  - used to carry EAP messages between the AP and the auth server
  - MS-MPPE-Recv-Key attribute is used to transport the session key from the auth server to the AP
  - RADIUS is mandated by WPA and optional for RSN

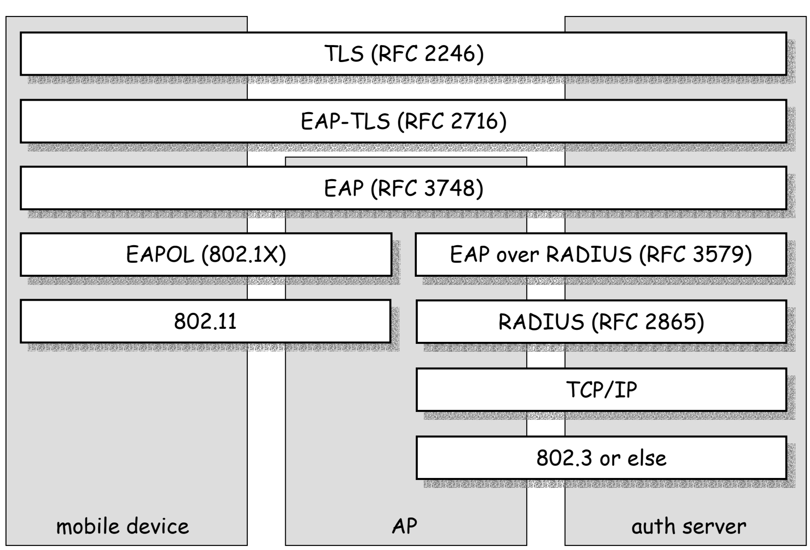
EAP in action



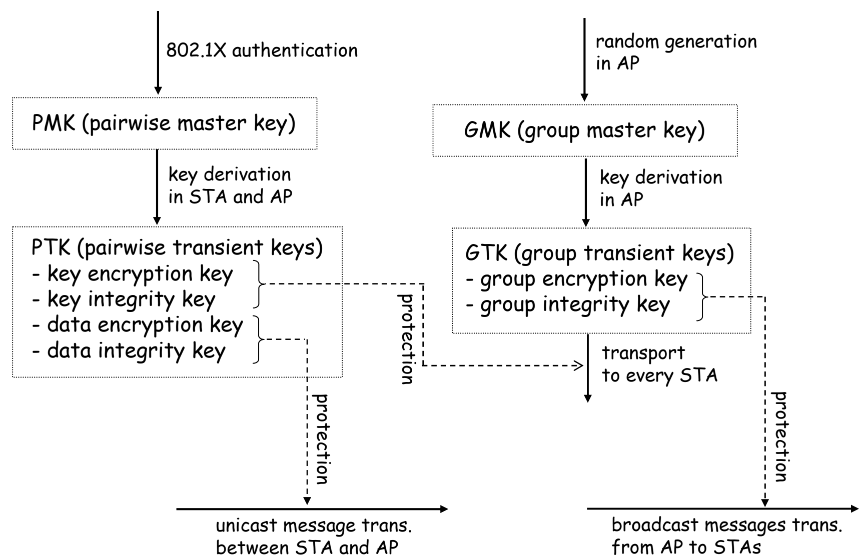
Protocols – LEAP, EAP-TLS, PEAP, EAP-SIM

- LEAP (Light EAP)
  - developed by Cisco
  - similar to MS-CHAP extended with session key transport
- EAP-TLS (TLS over EAP)
  - only the TLS Handshake Protocol is used
  - server and client authentication, generation of master secret
  - TLS maser secret becomes the session key
  - mandated by WPA, optional in RSN
- PEAP (Protected EAP)
  - phase 1: TLS Handshake without client authentication
  - phase 2: client authentication protected by the secure channel established in phase 1
- EAP-SIM
  - extended GSM authentication in WiFi context
  - protocol (simplified) :
    - STA → AP: EAP res ID (IMSI / pseudonym)
    - STA → AP: EAP res (nonce)
    - AP: [gets two auth triplets from the mobile operator's AuC]
    - AP → STA: EAP req (2*RAND | MIC_{2*Kc} | $\{\text{new pseudonym}\}_{2*Kc}$)
    - STA → AP: EAP res (2*SRES)
    - AP → STA: EAP success

Summary of the protocol architecture



Key hierarchies



Four-way handshake

- objective:
  - prove that AP also knows the PMK (result of authentication)
  - exchange random values to be used in the generation of PTK
- protocol:

AP : generate ANonce
AP → STA : ANonce | KeyReplayCtr
STA : generate SNonce and compute PTK
STA → AP : SNonce | KeyReplayCtr | MIC_{KIK}
AP : compute PTK, generate GTK, and verify MIC
AP → STA : ANonce | KeyReplayCtr+1 | $\{GTK\}_{KEK}$ | MIC_{KIK}
STA : verify MIC and install keys
STA → AP : KeyReplayCtr+1 | MIC_{KIK}
AP : verify MIC and install keys

PTK and GTK computation

- for TKIP

PRF-512(PMK,
"Pairwise key expansion",
MAC1 | MAC2 | Nonce1 | Nonce2) =
= KEK | KIK | DEK | DIK

PRF-256(GMK,
"Group key expansion",
MAC | GNonce) =
= GEK | GIK

- for AES-CCMP

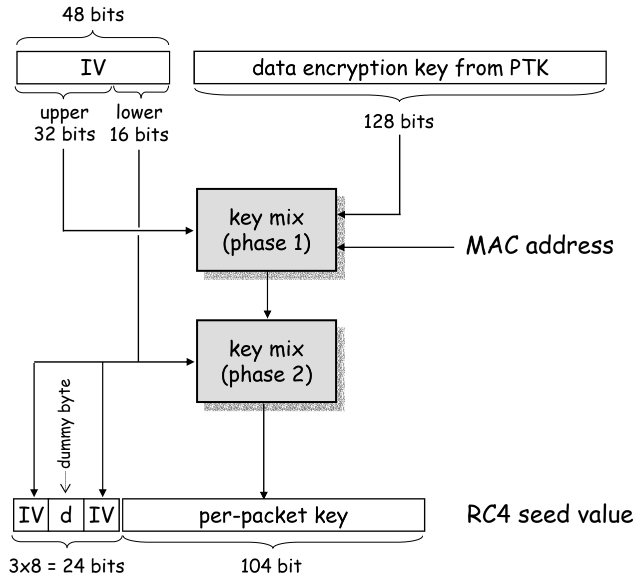
PRF-384(PMK,
"Pairwise key expansion",
MAC1 | MAC2 | Nonce1 | Nonce2) =
= KEK | KIK | DE&IK

PRF-128(GMK,
"Group key expansion",
MAC | GNonce) =
= GE&IK

TKIP

- runs on old hardware (supporting RC4), but …
- WEP weaknesses are corrected
  - new message integrity protection mechanism called Michael
    - MIC value is added at SDU level before fragmentation into PDUs
    - implemented in the device driver (in software)
  - use IV as replay counter
  - increase IV length to 48 bits in order to prevent IV reuse
  - per-packet keys to prevent attacks based on weak keys

TKIP – Generating RC4 keys



AES-CCMP

- CCMP means CTR mode and CBC-MAC
  - integrity protection is based on CBC-MAC (using AES)
  - encryption is based on CTR mode (using AES)
- CBC-MAC
  - CBC-MAC is computed over the MAC header, CCMP header, and the MPDU (fragmented data)
  - mutable fields are set to zero
  - input is padded with zeros if length is not multiple of 128 (bits)
  - CBC-MAC initial block:
    - flag (8)
    - priority (8)
    - source address (48)
    - packet number (48)
    - data length (16)
  - final 128-bit block of CBC encryption is truncated to (upper) 64 bits to get the CBC-MAC value
- CTR mode encryption
  - MPDU and CBC-MAC value is encrypted, MAC and CCMP headers are not
  - format of the counter is similar to the CBC-MAC initial block
    - "data length" is replaced by "counter"
    - counter is initialized with 1 and incremented after each encrypted block

Summary

- security has always been considered important for WiFi
- early solution was based on WEP
  - seriously flawed
  - not recommended to use
- the new security standard for WiFi is 802.11i
  - access control model is based on 802.1X
  - flexible authentication based on EAP and upper layer authentication protocols (e.g., TLS, GSM authentication)
  - improved key management
  - TKIP
    - uses RC4 → runs on old hardware
    - corrects WEP's flaws
    - mandatory in WPA, optional in RSN (WPA2)
  - AES-CCMP
    - uses AES in CCMP mode (CTR mode and CBC-MAC)
    - needs new hardware that supports AES

Recommended readings

- W. Arbaugh, N. Shankar, J. Wan, K. Zhang. Your 802.11 network has no clothes. *IEEE Wireless Communications Magazine,* 9(6):44-51, 2002.
- N. Borisov, I. Goldberg, D. Wagner. Intercepting mobile communications: the insecurity of 802.11. *Proceedings of the 7th ACM Conference on Mobile Computing and Networking,* 2001.
- B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748. 2004.
- **J. Edney, W. Arbaugh. *Real 802.11 Security: WiFi Protected Access and 802.11i.* Addison-Wesley, 2004.**
- S. Fluhrer, I. Mantin, A. Shamir. Weaknesses in the key scheduling algorithm of RC4. Proceedings of the 8th Workshop on Selected Areas in Cryptography. 2001.
- B. Aboba, P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support for Extensible Authentication Protocol (EAP), RFC 3579, 2003.
- J. Walker. Unsafe at any key size: An analysis of the WEP encapsulation. *IEEE 802.11-00/362*, 2000.
- Wi-Fi Alliance. Wi-Fi Protected Access. http://www.wi-fi.org/white_papers/whitepaper-042903-wpa/
- IEEE Std 802.1X-2001. IEEE Standard: Port-based Network Access Control, 2001.
- IEEE Std 802.11. IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- IEEE Std 802.11i. IEEE Standard Amendment 6: Medium Access Control (MAC) Security Enhancements, 2004.