WiFi Security

Levente Buttyán
CrySyS Lab, BME
buttyan@crysys.hu

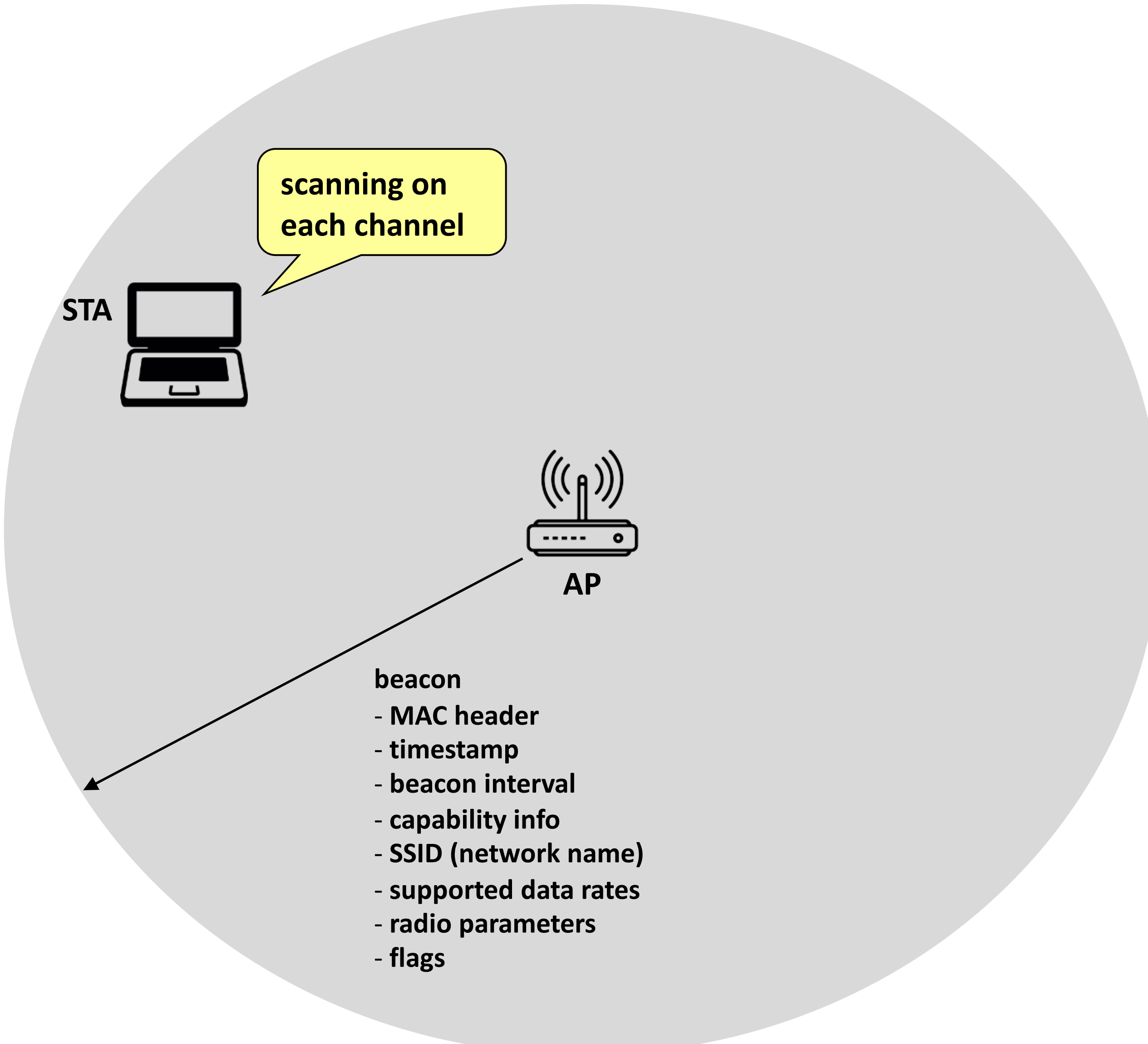
Security problems in wireless networks

- 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.)

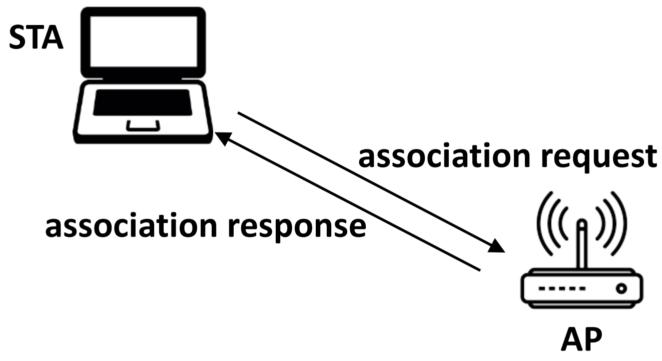
- broadcast communications
  - wireless usually means radio, which has a broadcast nature
  - transmissions can be overheard by anyone in range
  - anyone can generate transmissions
    - which will be received by other devices in range
    - which will interfere with other nearby transmissions and may 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

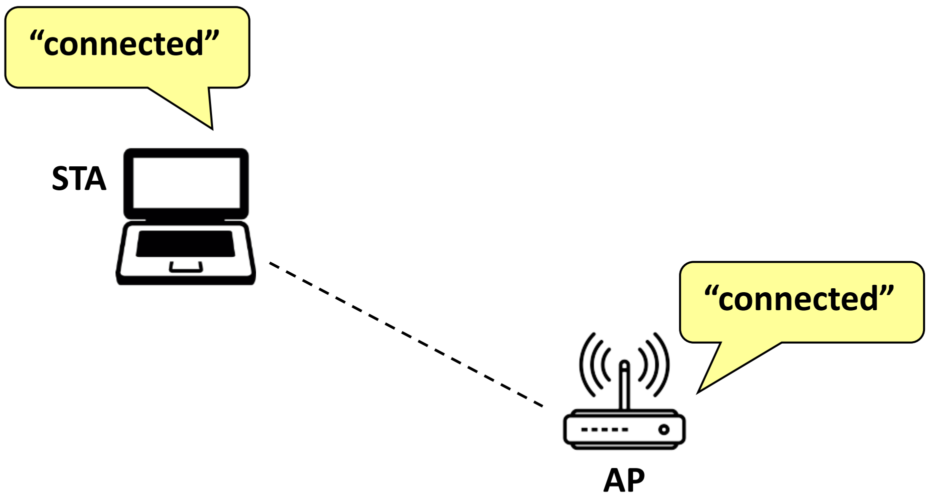
Brief reminder on the operation of WiFi



Brief reminder on the operation of WiFi



Brief reminder on the operation of WiFi



SSID-based access control

- SSID = Service Set IDentifier (network name)
  - a 32-character identifier
  - intended to differentiate one WLAN from another

- access to the network is sometimes prevented by not advertising the SSID publicly
  - only devices that know the “secret” SSID can connect to the network

- unfortunately, the SSID can be sniffed, and hence, this mechanism does not provide really secure access control
  - when a wireless station sends an association request to an access point, it includes the SSID of the network it wishes to associate with
  - an attacker can sniff this request and obtain the ”secret” SSID

MAC filtering-based access control

- MAC address filtering
  - only devices with certain MAC addresses are allowed to associate
  - needs pre-registration of all allowed devices at the AP
- unfortunately, MAC addresses can be sniffed and forged
  - sniffing
    - » MAC address is sent in clear in each packet
    - » put your WLAN adapter card in promiscuous mode (accepts all packets)
    - » eavesdrop the traffic and find out which MAC addresses are accepted
  - forging
    - » MAC address of certain WLAN adapter cards can be set by the user
    - » example:

```
# ifconfig ath0 hw ether <mac address of C>
```

WEP – Wired Equivalent Privacy

- part of the original IEEE 802.11 specification

- goal:
  - make the WiFi network *at least as secure as a wired LAN* (that has no particular protection mechanisms)

- services:
  - access control to the network
  - message confidentiality
  - message authenticity/integrity

- notes:
  - WEP has never intended to achieve strong security
  - at the end, it has **achieved no security at all due to bad design**

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)

  STA → AP: authenticate response (encrypted r)

  AP → STA: authenticate success/failure

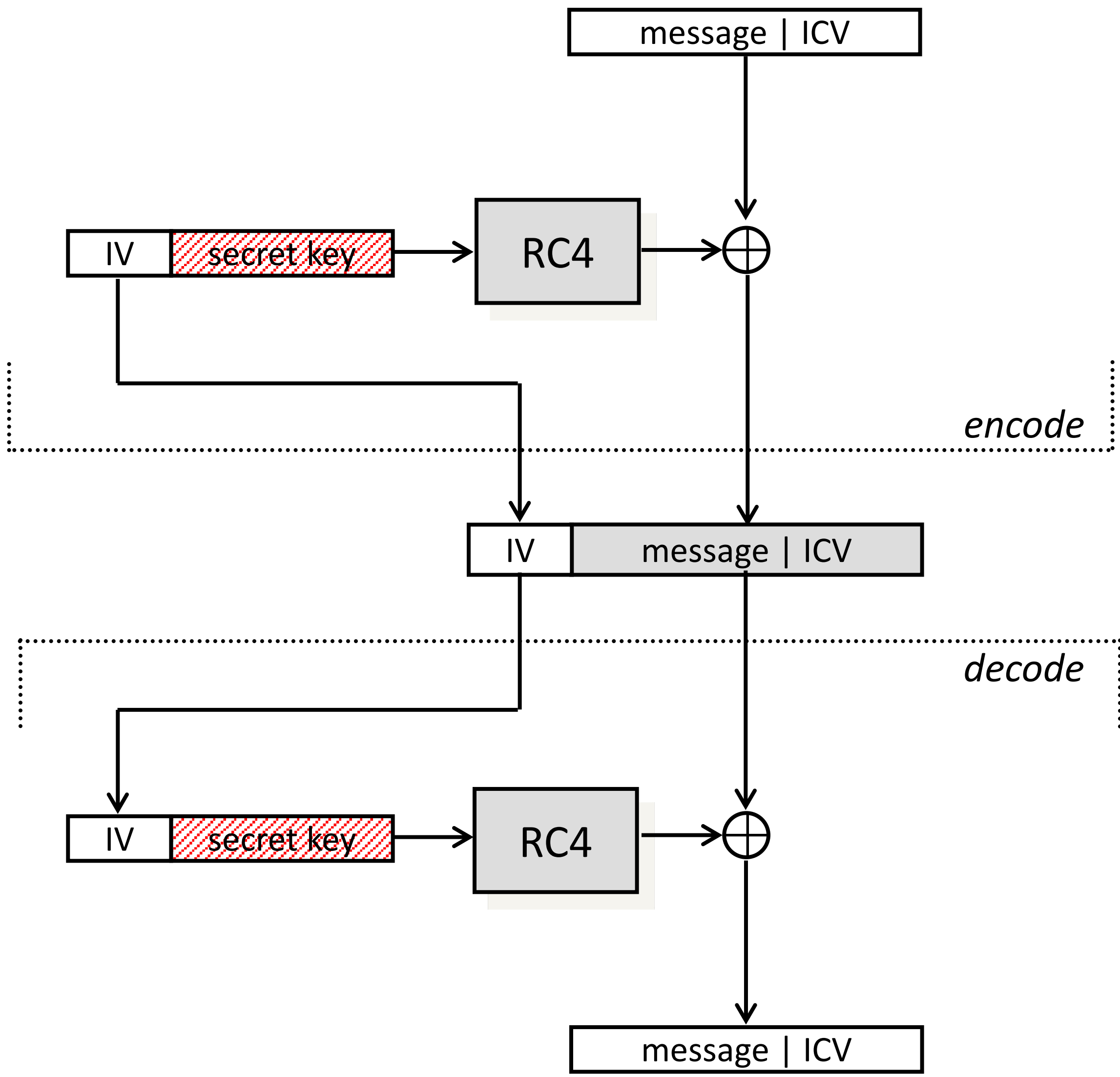
- once authenticated, the STA can send an association request, and the AP will respond with an association response
- if authentication fails, no association is possible

WEP – Message confidentiality and integrity

- WEP encryption is based on the RC4 stream cipher
  - it is essential that each message is encrypted with a different key stream
  - the RC4 cipher is initialized with a shared secret key and an IV (initial value)
    - shared secret key (40 or 104 bits) remains unchanged (static WEP key)
    - 24-bit IV is changed for every message sent
  - RC4 produces a pseudo-random byte sequence (key stream), which is XORed to the message
  - reception is analogous

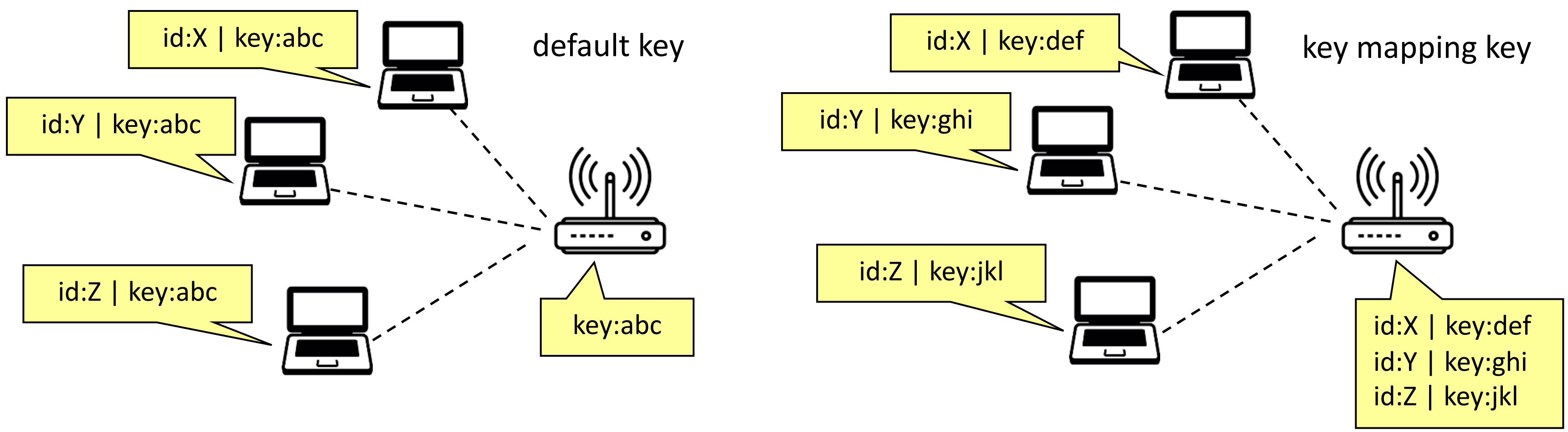
- WEP integrity protection is based on an encrypted CRC value
  - CRC of plaintext message is computed (called Integrity Check Value - ICV)
  - ICV is appended to the message
  - the message and the ICV are encrypted together as described above

WEP – Message protection illustrated



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)

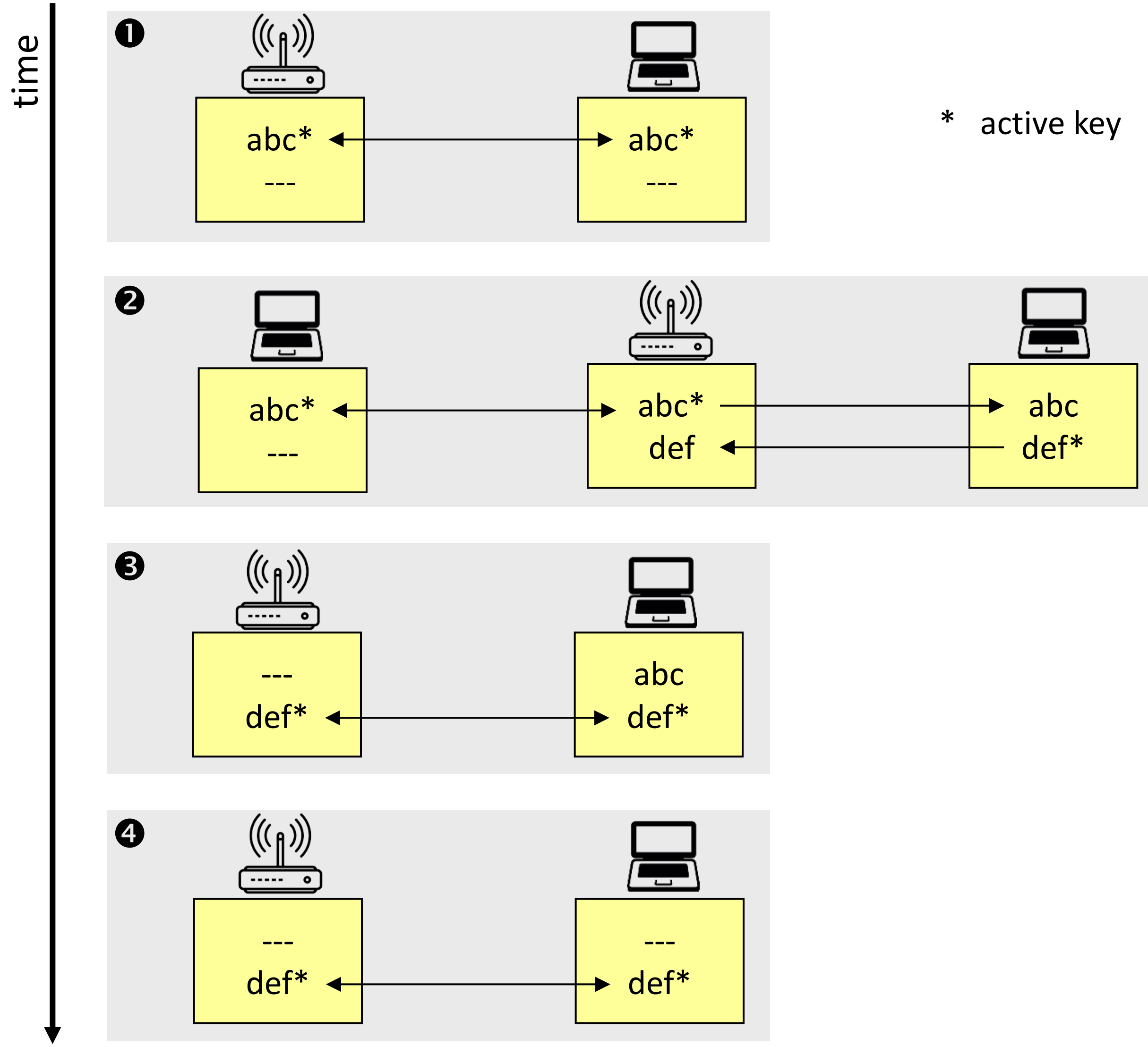


- in practice, often only default keys are supported
  - the default key is manually installed in every STA and the AP
  - each STA uses the same shared secret key → in principle, STAs can decrypt each other's messages

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



* active key

WEP flaws – Auth and access control

- authentication is one-way only
  - AP is not authenticated to STA
  - STA may associate to a rogue AP which can perform a Man-in-the-Middle attack
- 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 can 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 – Auth and access control

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

  …

  AP → STA: r

  STA → AP: IV | r + z

  …

  where z is a 128 bit RC4 output on IV and the shared WEP key

- an attacker can compute r + (r + z) = z

- she can use z (and the same IV) to impersonate STA later:

  …

  AP → attacker: r’

  attacker → AP: IV | r’ + z

  …

WEP flaws – Integrity and replay

- there's no replay protection at all
  - IV is not mandated to be incremented after each message
  - receiver is not mandated to check the freshness of received IVs

- attacker can manipulate messages despite the ICV mechanism and encryption
  - CRC is a linear function wrt to XOR:

$$CRC(X + Y) = CRC(X) + CRC(Y)$$

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

$$((M \mid CRC(M)) + Z) + (\Delta M \mid CRC(\Delta M)) =$$
$$((M + \Delta M) \mid (CRC(M) + CRC(\Delta M))) + Z =$$
$$((M + \Delta M) \mid CRC(M + \Delta M)) + Z$$

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 is capable for transmitting thousands of packets per second → IV space is used up in a few hours
  - in many implementations IVs are initialized with 0 on startup and then incremented by one after each message
    - 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
- exploiting information leakage by CRC verification
  - Chop-chop attack by KoreK
- exploiting weaknesses in the RC4 cipher
  - Fluhrer-Mantin-Shamir attack

Chopchop attack (KoreK)

- allows an attacker to interactively decrypt the last m bytes of the plaintext of a WEP encrypted packet by sending 128m crafted packets on average to the network, and observing if they are accepted or not (CRC is correct or not)
- does not reveal the WEP key!
- not based on any special properties of the RC4 stream cipher (protocol flaw!)

Chopchop attack – background

- every binary vector can be represented as a binary polynomial
  - e.g., 10010 → $1x^4 + 0x^3 + 0x^2 + 1x + 0 = x^4 + x$ (+ is the XOR operation)
- arithmetics over polynomials
  - polynomials can be added, multiplied, and divided with other polynomials
    - » e.g., $(x^2+1) / (x+1) = (x+1)$ (because $(x+1)(x+1) = x^2 + x + x + 1 = x^2+1$)
  - modulo with respect to a divisor can be defined similar to integers
    - » e.g., $(x^2+x+1) \bmod (x+1) = 1$ (becasue $x^2+x+1 = x(x+1) + 1$)
- CRC verification mechanism in WEP
  - (message|ICV) is represented as a binary polynomial P
  - if $P \bmod R_{CRC} = P_{ONE}$, then the message is accepted
    - » R_{CRC} is a given CRC polynomial (e.g., $x^{16} + x^{12} + x^5 + 1$)
    - » P_{ONE} is the polynomial with degree $\deg(R_{CRC})$-1 and all coefficients equal to one (i.e., P_{ONE} represents the all 1 vector 111...1)

Chopchop attack – background

- we can write P as $Qx^8 + L$, where
  - Q represents the one-byte shortened packet
  - L represents the last byte

- if P verifies correctly, then how do we need to modify Q such that it verifies correctly too?

$$P \bmod R_{CRC} = P_{ONE} = P_{ONE} \bmod R_{CRC}$$
$$(Qx^8 + L) \bmod R_{CRC} = Qx^8 \bmod R_{CRC} + L \bmod R_{CRC} = P_{ONE} \bmod R_{CRC}$$
$$Qx^8 \bmod R_{CRC} = (L + P_{ONE}) \bmod R_{CRC}$$
$$Q \bmod R_{CRC} = (L + P_{ONE})(x^8)^{-1} \bmod R_{CRC}$$

- let ΔQ be $P_{ONE} + (L + P_{ONE})(x^8)^{-1}$

$$(Q + \Delta Q) \bmod R_{CRC} = Q \bmod R_{CRC} + \Delta Q \bmod R_{CRC} =$$
$$= (L + P_{ONE})(x^8)^{-1} \bmod R_{CRC} + P_{ONE} + (L + P_{ONE})(x^8)^{-1} \bmod R_{CRC} = P_{ONE}$$

- $Q+\Delta Q$ verifies correctly and ΔQ depends only on L

Chopchop attack – the full monty

- eavesdrop a WEP encrypted packet P + Z
  - we know that P verifies correctly, but we don’t know P
- guess the last byte L of P
- chop the last byte of the encrypted packet and XOR in ΔQ to get (Q + Z’) + ΔQ = (Q + ΔQ) + Z’
- send (Q + ΔQ) + Z’ to the AP and observe if it is accepted
  - send the packet from a station not yet associated to the AP
  - if the packet is correct, the AP will send a message telling the station that it needs to rejoin the network, otherwise the packet is discarded
- if successful, you have the right value for the last byte L
- if unsuccessful, try another candidate for L
  - there are only 256 possibilities!
- **on average, after 128 trials you have the correct value for L**
- repeat the procedure by chopping the last byte of (Q + ΔQ)
- …

Lessons learned

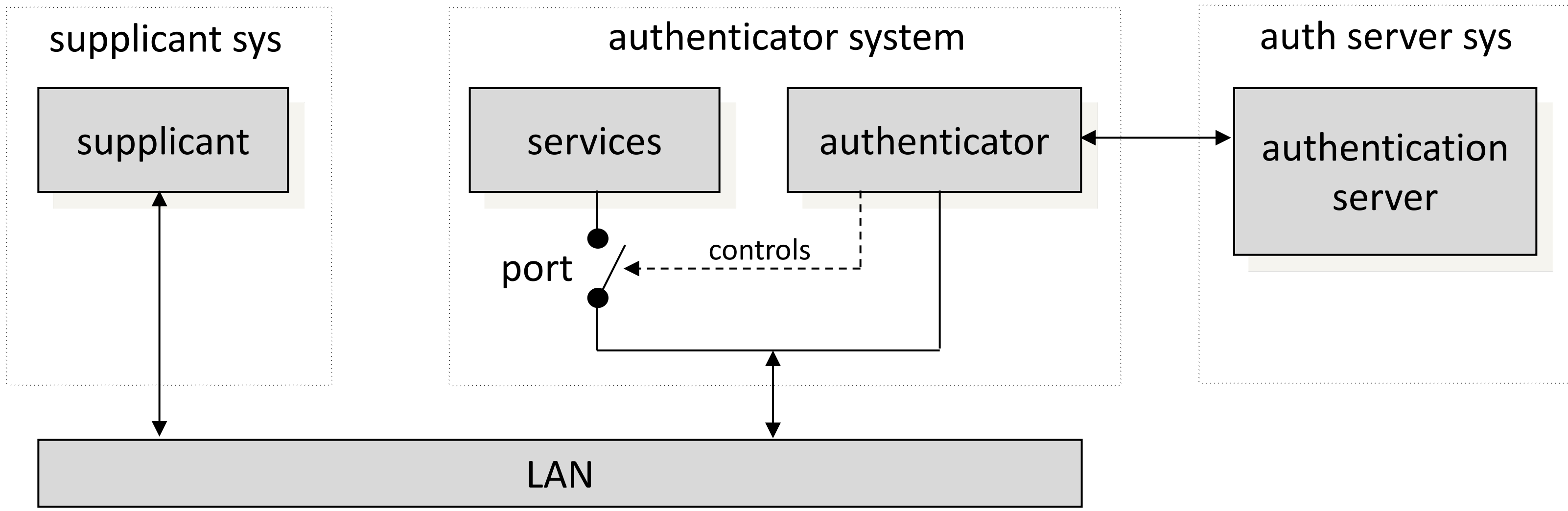
- engineering security protocols is a risky business
- you may combine otherwise strong building blocks in a wrong way and obtain an insecure system at the end
  - example:
    - stream ciphers alone are OK
    - challenge-response protocols for entity authentication are OK
    - but they shouldn’t be combined in a way done in WEP
  - example:
    - encrypting a message digest to obtain an ICV may be acceptable
    - but it doesn’t work if the message digest function is linear wrt to the encryption function

Overview of 802.11i

- after the collapse of WEP, IEEE started to develop a new security architecture → 802.11i (now integrated in 802.11)

- 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
  - replay protection is added
  - encryption function is improved

802.1X authentication model



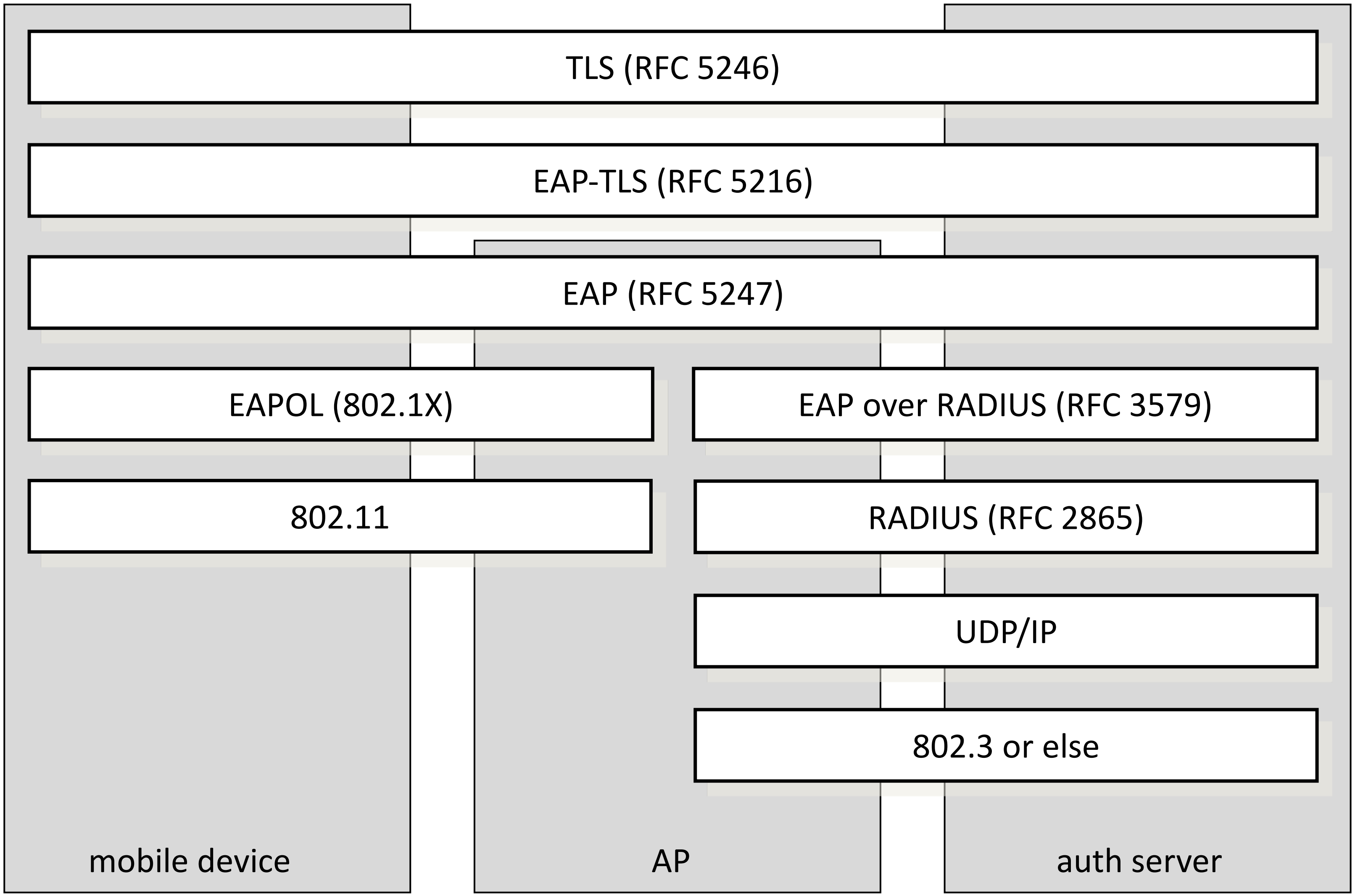
- the supplicant requests access to the services (wants to connect to the network)
- the authenticator controls access to the services (controls the state of a port)
- the authentication server authorizes access to the services
  - the supplicant authenticates itself to the authentication server
  - 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 remote 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

Protocol stack overview



EAP, EAPOL, RADIUS

- EAP (Extensible Authentication Protocol) [RFC 5247]
  - 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 authentication server to the supplicant
    - EAP response – carries messages from the supplicant to the authentication server
    - 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 WPA2

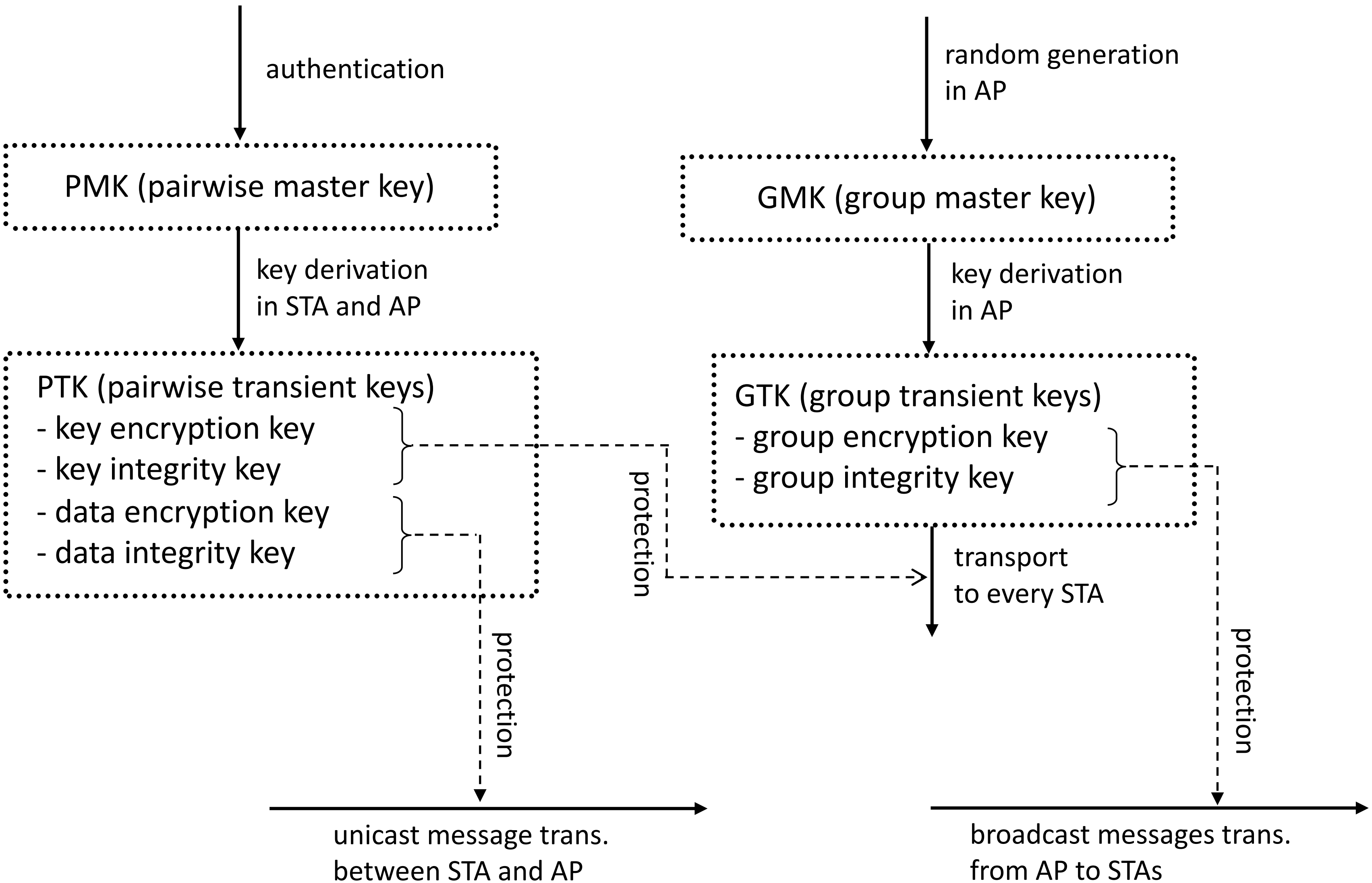
Authentication protocols supported

- EAP-TLS (TLS over EAP) [RFC 5216]
  - only the TLS Handshake Protocol is used
  - server and client authentication, generation of master secret
  - TLS master secret becomes the session key
  - mandated by WPA, optional in WPA2

- EAP-TTLS (Tunneled TLS over EAP) [RFC 5281]
  - phase 1: TLS Handshake possibly without client authentication
  - phase 2: legacy client authentication (e.g., password based) protected by the secure tunnel established in phase 1
    - » eavesdropping and man-in-the-middle attacks are prevented
    - » privacy is improved (user name is also encrypted)

- EAP-PSK, EAP-FAST, EAP-PEAP, EAP-SIM, EAP-AKA, …

Key hierarchy overview



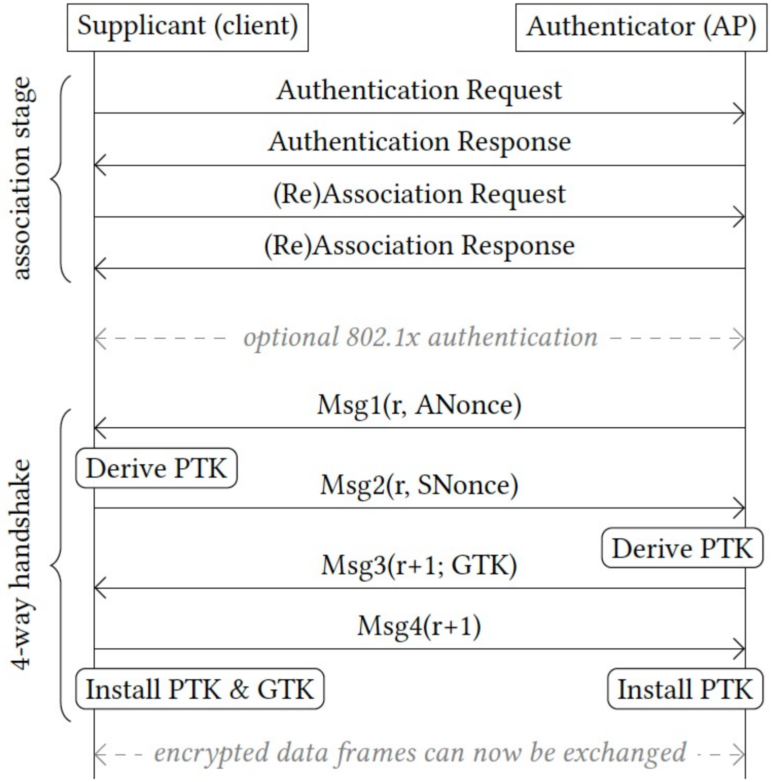
WPA and WPA2

- WPA (WiFi protected access)
  - industrial name for 802.11i TKIP (Temporal Key Integrity Protocol)
  - runs on old hardware (supporting RC4), but …
  - WEP weaknesses are corrected
    - IV is used as replay counter too
    - IV length is increased to 48 bits in order to prevent IV reuse
    - per-packet keys are used to prevent attacks similar to that of the FMS attack
    - integrity protection is based on Michael (old CRC is still used → chopchop)
- WPA2
  - industrial name for 802.11i AES-CCMP (Counter mode encryption and CBC-MAC Protocol)
  - integrity protection and encryption is based on AES (in CCM mode)
  - nice solution, but needs new hardware

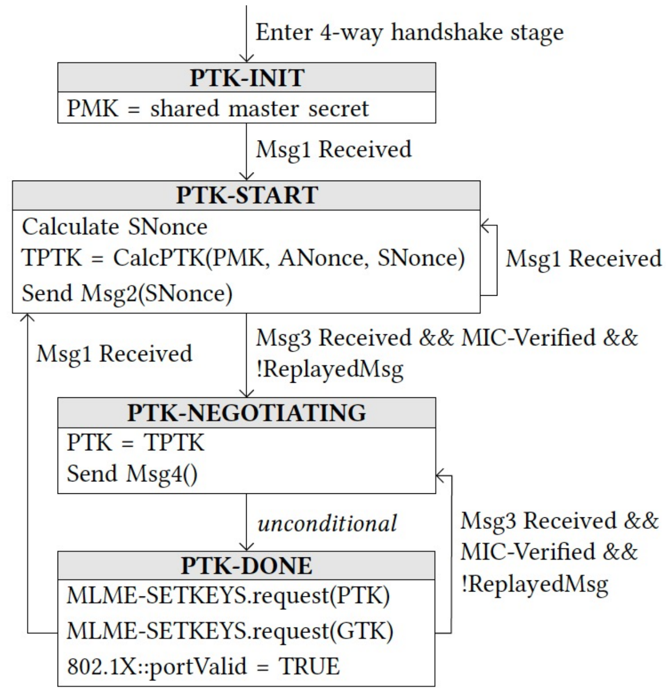
4-way handshake and its attack

- PTK is derived from PMK and GTK is distributed to supplicant stations via the **4-way handshake** protocol
- It has been formally proven to be secure, so it was not under strong investigation
- A devastating attack was found and published in 2017
  - The attacker can force re-installation of an already used PTK
  - As a consequence, counters are reset and the same counter values are used again with the same keys --» this is bad in case of CTR mode!!!
  - In some implementations, memory that holds the PTK is filled with zeros once the PTK is installed; in these implementations, forced re-installation of the PTK deploys the all-zero keys
    - » 31% of all Android devices were affected in this way!!!
- Interestingly, the formal proof is still valid
  - PTK is not leaked by the attack (in general)
  - the problem is that the proof does not model key installation

The 4-way handshake protocol



protocol

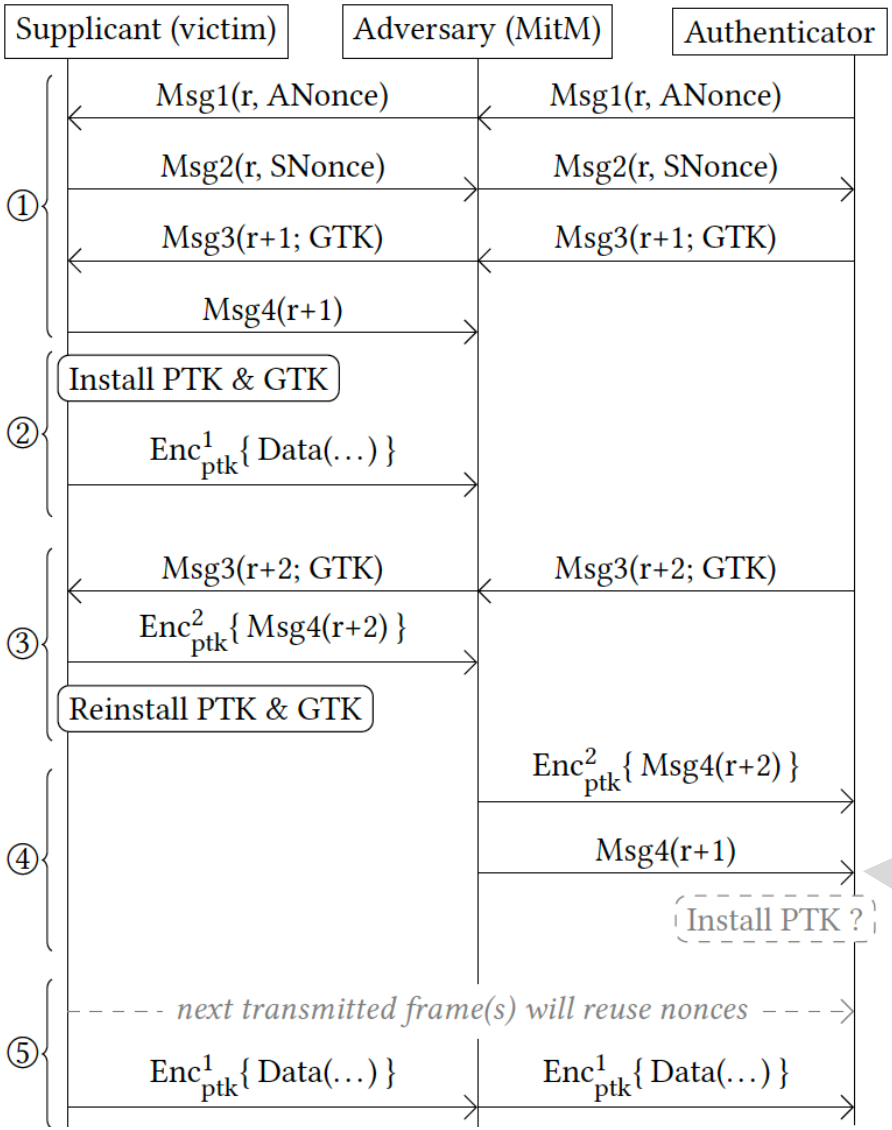


supplicant state machine

Attacking the 4-way handshake

- the supplicant accepts retransmissions of message 3, even when it is in the PTK-DONE state
  - this is to handle the situation when message 4 is lost and the AP re-sends message 3
- a retransmitted message 3 re-installs the PTK (and the GTK)
  - with all the negative conseqiences of resetting counters
- the attack requires a Man-in-the-Middle position
  - retransmissions of message 3 is triggered by preventing message 4 from arriving at the AP

Attack illustrated



The AP may accept any replay counter that was used in the handshake, not only the latest one:

"On reception of message 4, the Authenticator verifies that the Key Replay Counter field value is one that it used on this 4-way handshake."

In practice, many APs indeed accept an older replay counter if no message with that counter was received before.

Summary on WiFi security

- security has always been considered important for WiFi
- early solution was based on WEP
  - seriously flawed !
  - good example for bad security design
- WPA and WPA2
  - access control model is based on 802.1X
  - flexible authentication based on EAP and upper layer authentication protocols (e.g., TLS, 3G authentication)
  - improved key management
  - WPA (TKIP)
    - uses RC4 → runs on old hardware, but corrects (most of) WEP's flaws
    - still vulnerable to the chopchop attack
  - WPA2 (AES-CCMP)
    - uses AES in CCM mode (an authenticated encryption mode)
    - needs new hardware that supports AES

Further readings

- 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.
- 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.
- M. Beck, E. Tews, Practical attacks against WEP and WPA, Proceedings of the ACM Conference on Wireless Network Security, March 16-18, 2009.
- M. Vanhoef, F. Piessens, Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, ACM Conference on Computer and Communications Security, 2017. (https://www.krackattacks.com/)

Control questions

- What are the main security problems in wireless networks?
- Why undisclosed SSIDs and MAC filtering don’t provide security for WiFi?
- What are the main security objectives of WEP? How are they attempted to be achieved? Explain the details of WEP!
- Why does a group key need to be changed when someone leaves the group? How default WEP keys are updated in WEP?
- What are the flaws in WEP? Explain the details!
- How does authentication work in WPA and WPA2? What are the main protocols involved and how are they stacked on each other?
- What keys are derived after authentication?
- How are message confidentiality and integrity provided in WPA?
- How are weaknesses of WEP addressed in WPA? What weaknesses do still remain?
- How are message confidentiality and integrity provided in WPA2?
- What attacks against WPA2 do you know? Explain their main idea briefly!