WIFI SECURITY

ENCRYPTION REVISITED

We have covered encryption in chars and blocks.

But what about the security, if two k-bit blocks are encrypted the same?

We need some randomness, so identical block encrypt differently.

CIPHER BLOCK CHAINING

To avoid two k-bit blocks are encrypted the same, we need some randomness, so identical block encrypt differently.

- Let **m(i)** denote the i^{th} plaintext block
- Let **c(i)** denote the i^{th} ciphertext block
- Let **a ⊕ b** denote XOR of a and b
- Let K_S denote Encryption algorihm with key S

CIPHER BLOCK CHAINING

Basic idea: Sender creates a random k-bit number **r(i)** for the **i** th block and calculates: $c(i) = K_S(m(i) \oplus r(i))$

The sender then sendt: **c(1),r(1),c(2),r(2),c(3),r(3)…**

Receiver can now calculate: $m(i) = K_S(c(i) \oplus r(i))$

Althrough **r(i)** is sent in the plain, Trudy cannot obtain **m(i)** as she does not know K_S

Problem: Must transmit twice the number of bits.

CIPHER BLOCK CHAINING

- Basic idea: Cipher Block Chaining

  Send only one random value along with the very first message, and then have the sender and receiver use the *computed code blocks* in place of the subsequent random number.

CIPHER BLOCK CHAINING

CBC operates as follows

- Before encrypting the message, the sender generates a random k-bit string, the **Initialization Vector (IV)**. The sender sends the IV in cleartext. Let **c(0)** denote the IV.
- For the first block, sender calculates **m(1) ⊕ c(0)** and encrypts with the key: **$c(1) = K_S(m(1) \oplus c(0))$**
- For the **i^{th}** block, the sender generates the **i^{th}** ciphertext block from **$c(i) = K_S(m(i) \oplus c(i-1))$**

CIPHER BLOCK CHAINING

Consequences:

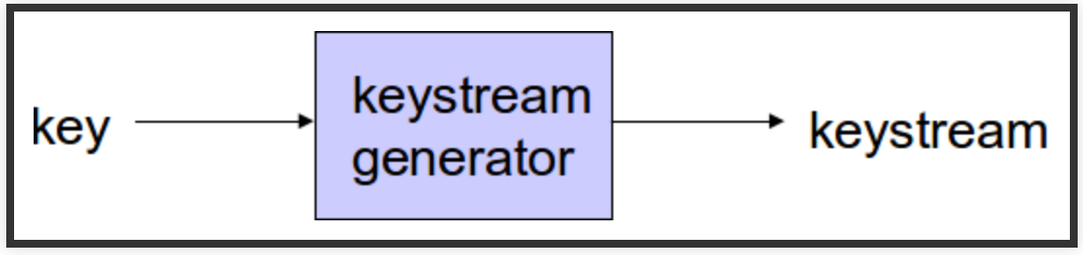
- Receiver can still decrypt the original message: **$m(i) = K_S(c(i) \oplus c(i-1))$**
- Even if two cleartext blocks are identical, the corresponding ciphertexts will (almost allways) be different.
- Only one overhead block
- **Issue:** If one block is missing, the blocks following that cannot be decrypted.

SECURING WIRELESS LANS

WEP DESIGN GOALS

- Symmetric key crypto
  - Confidentiality
  - End host authorization
  - Data integrity
- Self-synchronizing: each packet separately encrypted
  - given encrypted packet and key, can decrypt; can continue to decrypt packets when preceding packet was lost (unlike Cipher Block Chaining (CBC) in block ciphers)
- Efficient
  - implementable in hardware or software

SYMMETRIC STREAM CIPHERS



RC4 - RIVEST CIPHER 4

- RC4 generates a pseudorandom stream of bits (a keystream).
- RC4 was designed by Ron Rivest of RSA Security in 1987. Initially a trade secret.
- RC4 (also known as ARC4 or ARCFOUR meaning Alleged RC4) is the most widely used software stream cipher
- Simple and fast
- Vulnerable when the beginning of the output keystream is not discarded, or when nonrandom or related keys are used

RC4 - RIVEST CIPHER 4

- Makes use of a secret internal state which consists of two parts:
  - A permutation of all 256 possible bytes.
  - Two 8-bit index-pointers
- The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA).
- The stream of bits is generated using the pseudo-random generation algorithm (PRGA).

RC4 - RIVEST CIPHER 4

Notes:

- Variations exist
- Does suffer from some vulnerabilities

RC4 - RIVEST CIPHER 4

Key-scheduling algorithm (KSA)

```
for i from 0 to 255
  S[i] := i
endfor
j := 0
for i from 0 to 255
  j := (j + S[i] + key[i mod keylength]) mod 256
  swap values of S[i] and S[j]
endfor
```

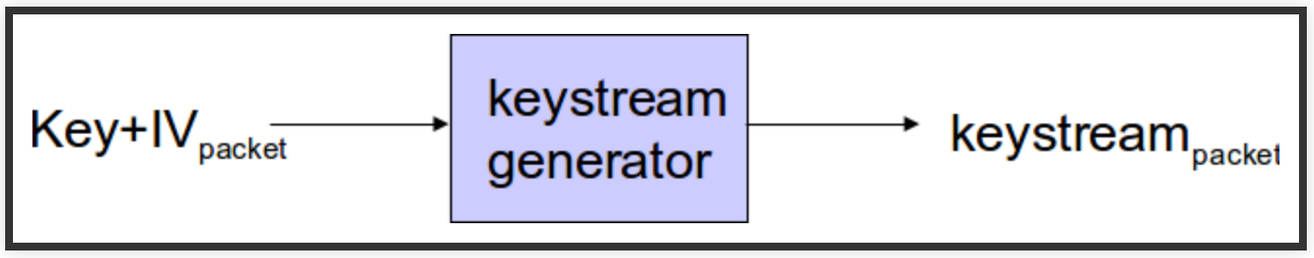
RC4 - RIVEST CIPHER 4

Pseudo-random generation algorithm (PRGA)

```
i := 0
j := 0
while GeneratingOutput:
  i := (i + 1) mod 256
  j := (j + S[i]) mod 256
  swap values of S[i] and S[j]
  K := S[(S[i] + S[j]) mod 256]
  output K
endwhile
```

STREAM CIPHER AND PACKET INDEPENDENCE

- Recall design goal: each packet separately encrypted
- if for frame n+1, use keystream from where we left off for frame n, then each frame is not separately encrypted
  - need to know where we left off for packet n
- WEP approach: initialize keystream with key + new IV for each packet:



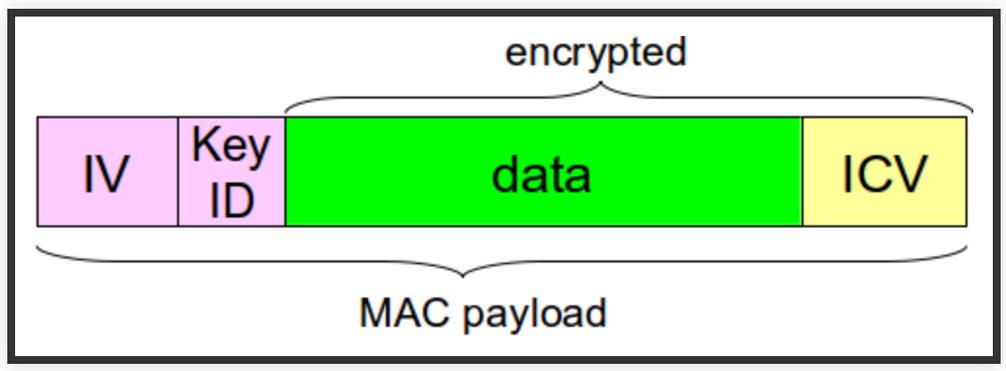
WEP ENCRYPTION

WEP ENCRYPTION

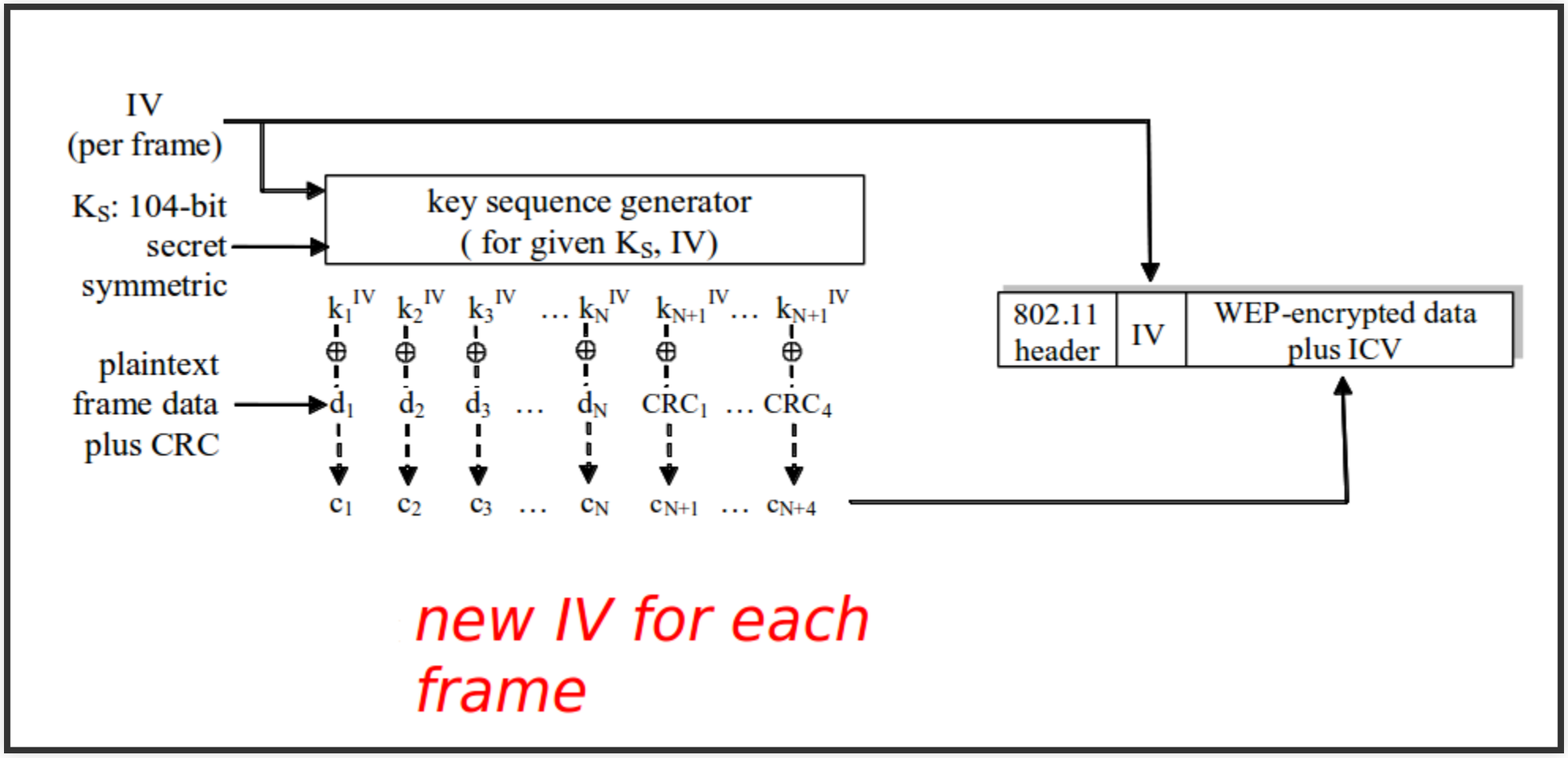
- sender calculates Integrity Check Value (ICV) over data
  - four-byte hash/CRC for data integrity
- each side has 104-bit shared key
- sender creates 24-bit initialization vector (IV), appends to key: gives 128-bit key
- sender also appends keyID (in 8-bit field)
- 128-bit key inputted into pseudo random number generator to get keystream

WEP ENCRYPTION

- data in frame + ICV is encrypted with RC4:
  - B bytes of keystream are XORed with bytes of data and ICV
  - IV and keyID are appended to encrypted data to create payload
  - payload inserted into 802.11 frame

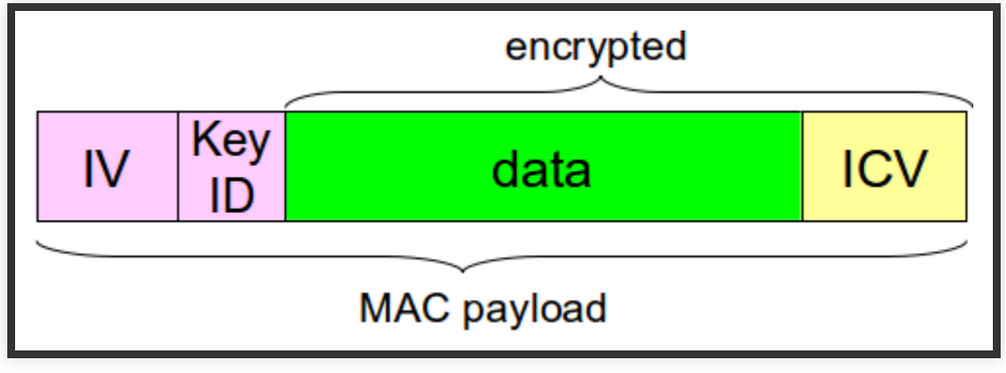


WEP ENCRYPTION



new IV for each frame

WEP DECRYPTION OVERVIEW

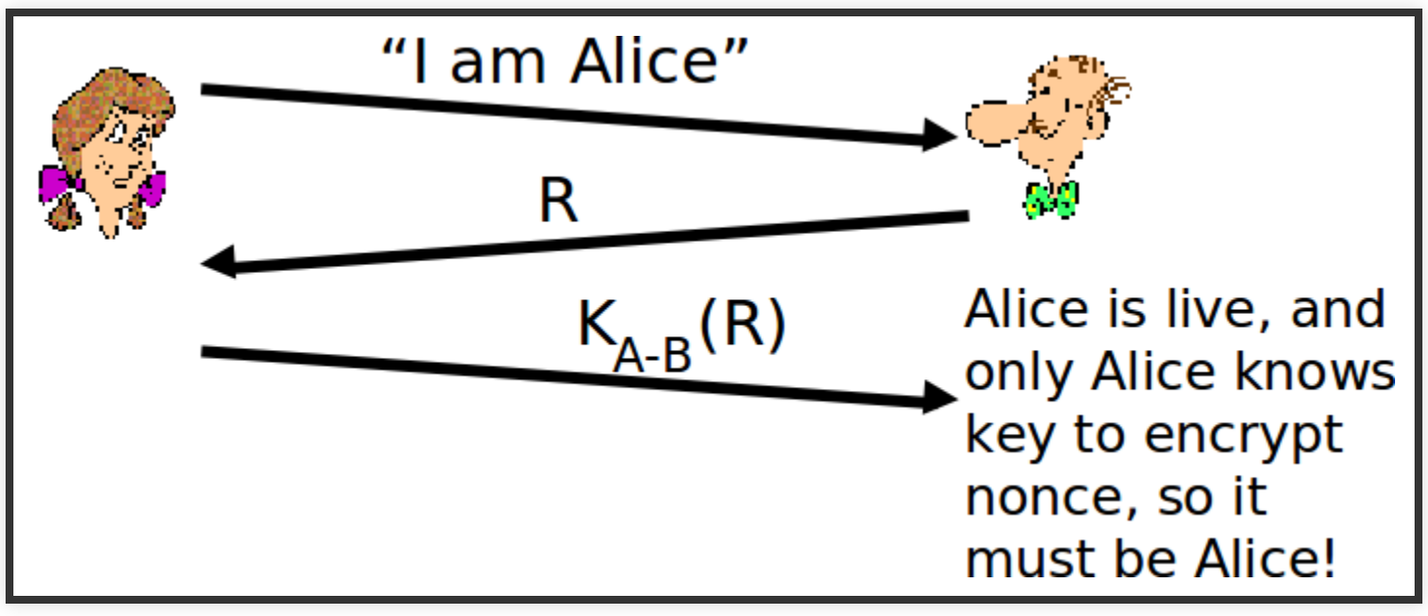


- receiver extracts IV
- inputs IV, shared secret key into pseudo random generator, gets keystream
- XORs keystream with encrypted data to decrypt data + ICV
- verifies integrity of data with ICV
  - note: message integrity approach used here is different from MAC (message authentication code) and signatures (using PKI).

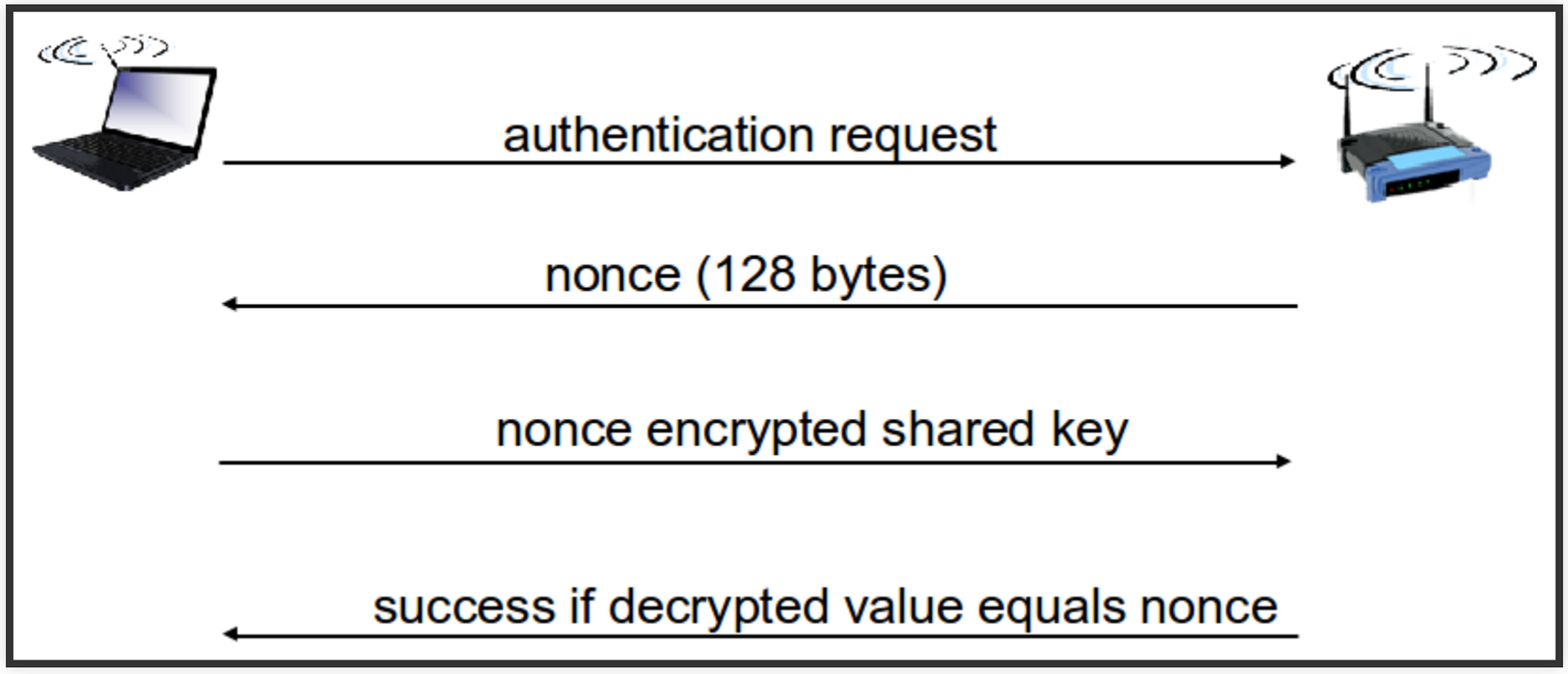
END-POINT AUTHENTICATION W/ NONCE

Nonce: number **R** used only once-in-a-lifetime

How to prove Alice "live": Bob sends Alice **nonce**, R. Alice must return R, encrypted with shared secret key



WEP AUTHENTICATION



Notes:

- not all APs do it, even if WEP is being used
- AP indicates if authentication is necessary in beacon frame
- done before association

BREAKING 802.11 WEP ENCRYPTION

SECURITY HOLE

- 24-bit IV, one IV per frame, → IV's eventually reused
- IV transmitted in plaintext → IV reuse detected

ATTACK

- Trudy causes Alice to encrypt known plaintext d_1 d_2 d_3 d_4 …
- Trudy sees: $c_i = d_i$ XOR k_i^{IV}
- Trudy knows c_i d_i, so can compute k_i^{IV}
- Trudy knows encrypting key sequence k_1^{IV} k_2^{IV} k_3^{IV} …
- Next time IV is used, Trudy can decrypt!

WPA2 - 802.11I

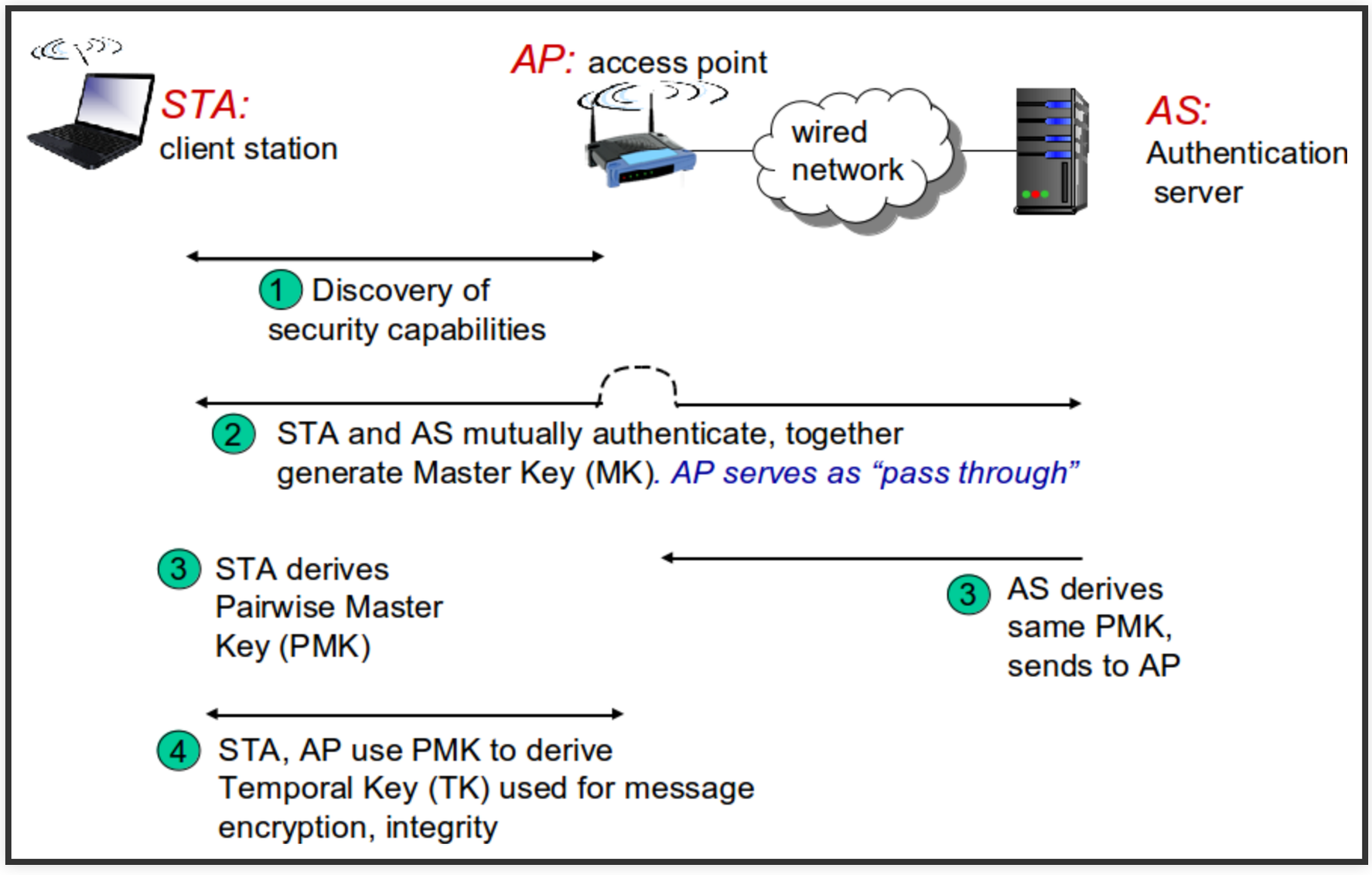
WPA2 802.11I: IMPROVED SECURITY

- Numerous (stronger) forms of encryption possible
- Provides key distribution
- Can use authentication server separate from access point

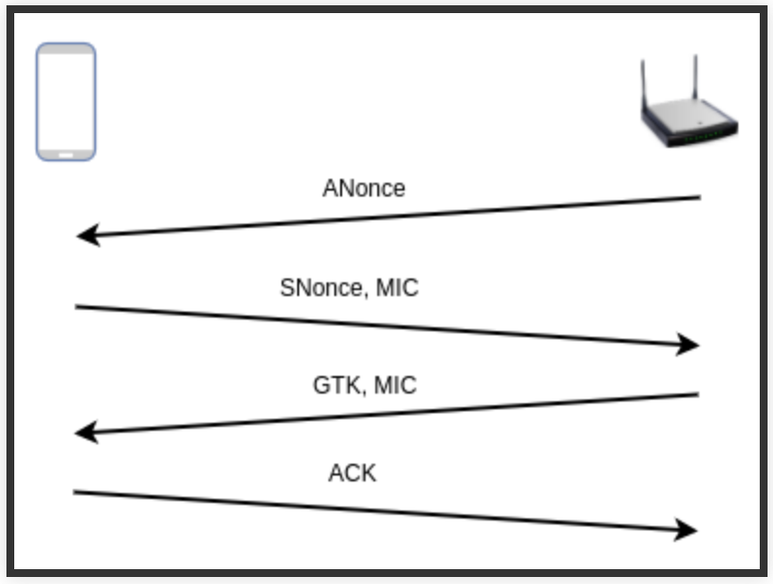
WPA2 - NOTES

- WPA implemented a subset of a draft of 802.11i.
- The full 802.11i is referred to as WPA2
  - Also called RSN (Robust Security Network).
- Advanced Encryption Standard (AES) block cipher
  - WEP and WPA use the RC4 stream cipher.

WPA2 - FOUR PHASES OF OPERATION

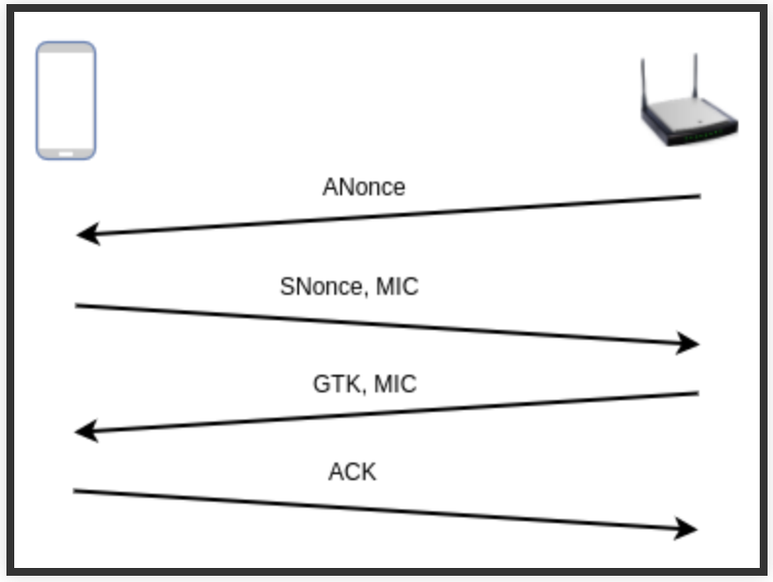


WPA2 - 4 WAY HANDSHAKE AND KEY GENERATION



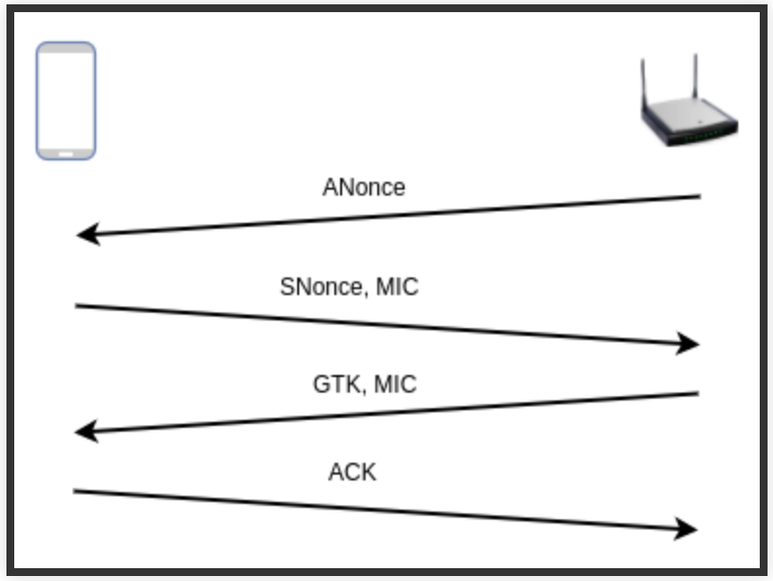
- AP Sends nonce, ANonce - random 128 bit

WPA2 - 4 WAY HANDSHAKE AND KEY GENERATION



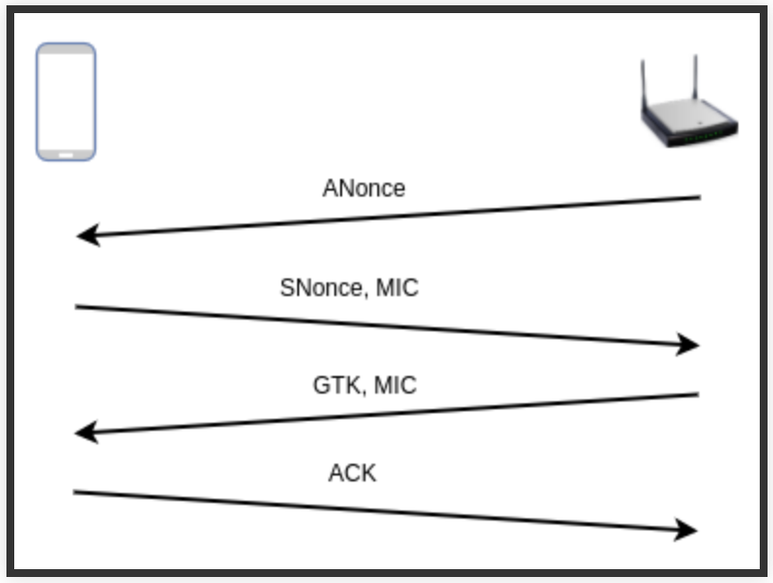
- Client makes random nonce SNonce
- Client uses ANonce, SNonce, AP MAC, Client MAC, WPA2 Password (PMK) to generate PTK (Pairwise Transient Key)
- MIC (Message Integrity Code) also include ANonce

WPA2 - 4 WAY HANDSHAKE AND KEY GENERATION



- AP uses ANonce, SNonce, AP MAC, Client MAC, WPA2 Password (PMK) to generate PTK.
- Send Group Temporal Key (GTK) for broadcast, and MIC

WPA2 - 4 WAY HANDSHAKE AND KEY GENERATION



- ACK of the keys

SECURITY PROBLEMS

KRACK ATTACK ON WPA2

Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2

Mathy Vanhoef
imec-DistriNet, KU Leuven
Mathy.Vanhoef@cs.kuleuven.be

Frank Piessens
imec-DistriNet, KU Leuven
Frank.Piessens@cs.kuleuven.be

ABSTRACT

We introduce the key reinstallation attack. This attack abuses design or implementation flaws in cryptographic protocols to reinstall an already-in-use key. This resets the key's associated parameters such as transmit nonces and receive replay counters. Several types of cryptographic Wi-Fi handshakes are affected by the attack.

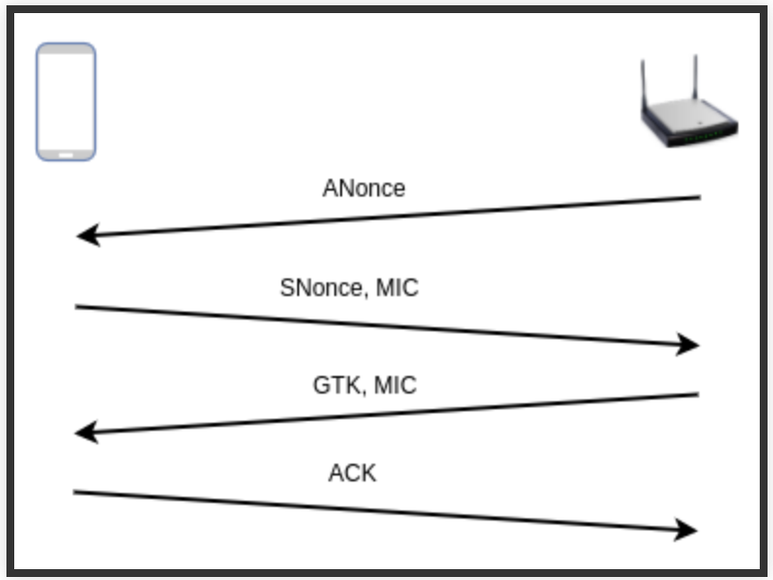
All protected Wi-Fi networks use the 4-way handshake to generate a fresh session key. So far, this 14-year-old handshake has remained free from attacks, and is even proven secure. However, we show that the 4-way handshake is vulnerable to a key reinstallation attack. Here, the adversary tricks a victim into reinstalling an already-in-use key. This is achieved by manipulating and replaying

work, we present design flaws in the 4-way handshake, and in related handshakes. Because we target these handshakes, both WPA- and WPA2-certified products are affected by our attacks.

The 4-way handshake provides mutual authentication and session key agreement. Together with (AES)-CCMP, a data-confidentiality and integrity protocol, it forms the foundation of the 802.11i amendment. Since its first introduction in 2003, under the name WPA, this core part of the 802.11i amendment has remained free from attacks. Indeed, the only currently known weaknesses of 802.11i are in (WPA-)TKIP [57, 66]. This data-confidentiality protocol was designed as a short-term solution to the broken WEP protocol. In other words, TKIP was never intended to be a long-

KRACK ATTACK ON WPA2

Remember the 4 way Handshake



KRACK ATTACK ON WPA2

- 3rd message can be resent multiple times, causing a reset of transmission nonces.
- As a result, same encryption key is used with with nonce values that have already been used in the past
  - Violates perfect forward secrecy
- The reset causes all encryption protocols of WPS2 to reuse keystream when encrypting packets.
  - Keystream will decrypt all packets with same nonce.

KRACK ATTACK ON WPA2

- https://www.krackattacks.com/
- https://papers.mathyvanhoef.com/ccs2017.pdf

WIFI PROTECTED SETUP

WiFi Protected Setup (WPS)

- The Wi-Fi Alliance in 2006
- Goal: Allow home users to set up Wi-Fi Protected Access easily
- Make it easy to add new device to existing network.

WIFI PROTECTED SETUP

- susceptible to a brute force attack
- allows an attacker to know when the first half of the 8 digit PIN is correct
- last digit of the PIN is known because it is a checksum for the PIN.
- The number of attempts goes from 10^8 to $10^4 + 10^3 = 11{,}000$ attempts in total.
- The lack of a proper lock out policy after a certain number of failed attempts to guess the PIN on many wireless routers makes this brute force attack that much more feasible.
- Maybe not possible to turn off!

WIFI PROTECTED SETUP

- http://www.kb.cert.org/vuls/id/723755
- http://arstechnica.com/business/2012/01/hands-on-hacking-wifi-protected-setup-with-reaver/

WIFI PROTECTED SETUP


CHAHUA 茶花
兔形粘钩
ADHESIVE HOOK
No.2960
• 卡通粘钩系列 •
茶花
MAX LOAD
1kg

OPEN HOTSPOTS

With no login

- No guaranteed encryption - sniffing possible
- Everyone can see
  - What unencrypted web pages you're visiting
  - What you're typing into unencrypted web forms
  - Even see which encrypted websites you're connected to
    - First after connection will it be encrypted → Metadata

> Browser plugin: HTTPS Everywhere

SUMMARY

Or more a final remark

- Use WPA2, not WEP or WPA as both of the later are compromised

QUESTIONS