Advanced Wi-Fi Security Penetration Testing

Vivek Ramachandran

http://www.securitytube.net

vivek@securitytube.net

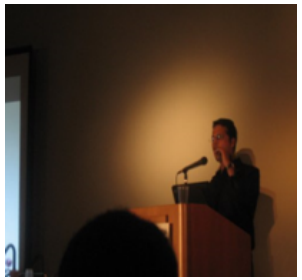
Please turn in your completed feedback form at the registration desk.

Vivek Ramachandran

B.Tech, ECE
IIT Guwahati

802.1x, Cat65k
Cisco Systems

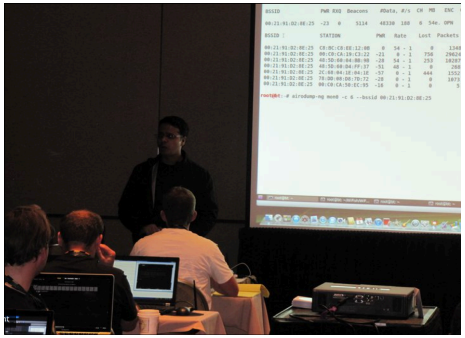
WEP Cloaking
Defcon 15



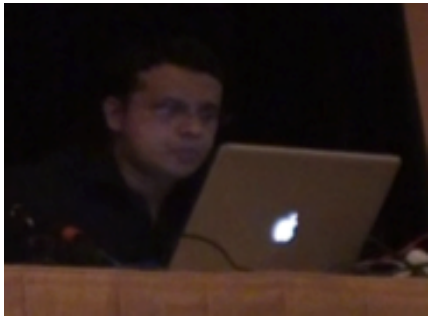
Caffe Latte Attack
Toorcon 9

Media Coverage
CBS5, BBC

Microsoft
Security Shootout



Trainer, 2011



Wi-Fi Malware, 2011

Security Research at Hacker Cons in 2011

August, Las Vegas

August, Las Vegas

Dec, Pune

Sept, Hungary

Sept, Belgium

Nov, Columbia

Dec, Abu Dhabi

October, Cochin

Why is Wireless Security Important?

- Seamless mobility
- Ubiquitous
- Mass adoption
- Integrated into all devices
  - Laptops
  - Phones
  - Embedded Devices
- Connects to Internet

Wireless Security Challenges

- How do you protect something you can't see? ☺
- Extends beyond boundary walls
- Mobile clients
- Difficult to locate attacker
- Passive attacks can be done from miles away

Wireless Gear

Victim

Attacker

Access Point

External Card

Smartphone

- 2 Laptops
- 1 Smartphone
- 1-2 Access Points
- 1-2 External Wireless Cards

External Wireless Card

- Alfa Networks AWUS036H USB based card
- Already integrated into Backtrack
- Allows for packet sniffing
- Allows for packet injection
- Maximum advertised output at 1 Watt
- We will use this in all our experiments
- Current Retail Price at $37 on Amazon http://www.amazon.com/Alfa-802-11b-Wireless-Original-9dBi/dp/B001O9X9EU

Software Setup

- Run Backtrack in VirtualBox
- Load the Lab Files on it

Understanding Wireless Sniffing

- Concept similar to wired side sniffing
- Put the wireless interface into “monitor” mode
  - Akin to wired side “promiscuous” mode
- On BT tools are inbuilt
- Will use Airmon-NG to put the card into monitor mode

Lab Session: Simple Sniffing

- Start sniffing on the air
- Use wireshark to see the packets

Surprise! Wi-Fi Sniffing is more complicated

- WLANs can operate in 3 different frequency ranges
  - 2.4GHz (802.11b/g/n)
  - 3.6GHz (802.11y)
  - 4.9/5.0GHz (802.11a/h/j/n)
- Each of these ranges is divided into multiple channels
- Every country has allowed channels, users and maximum power levels
- However, wireless card can be configured to disregard these policies

802.11b/g/n Channels

channel	frequency (MHz)	North America [3]	Japan[3]	Most of world[A] [3][4][5][6][7]
1	2412	Yes	Yes	Yes[D]
2	2417	Yes	Yes	Yes[D]
3	2422	Yes	Yes	Yes[D]
4	2427	Yes	Yes	Yes[D]
5	2432	Yes	Yes	Yes
6	2437	Yes	Yes	Yes
7	2442	Yes	Yes	Yes
8	2447	Yes	Yes	Yes
9	2452	Yes	Yes	Yes
10	2457	Yes	Yes	Yes
11	2462	Yes	Yes	Yes
12	2467	No[B]	Yes	Yes
13	2472	No[B]	Yes	Yes
14	2484	No	11b only[C]	No

- Source: Wikipedia

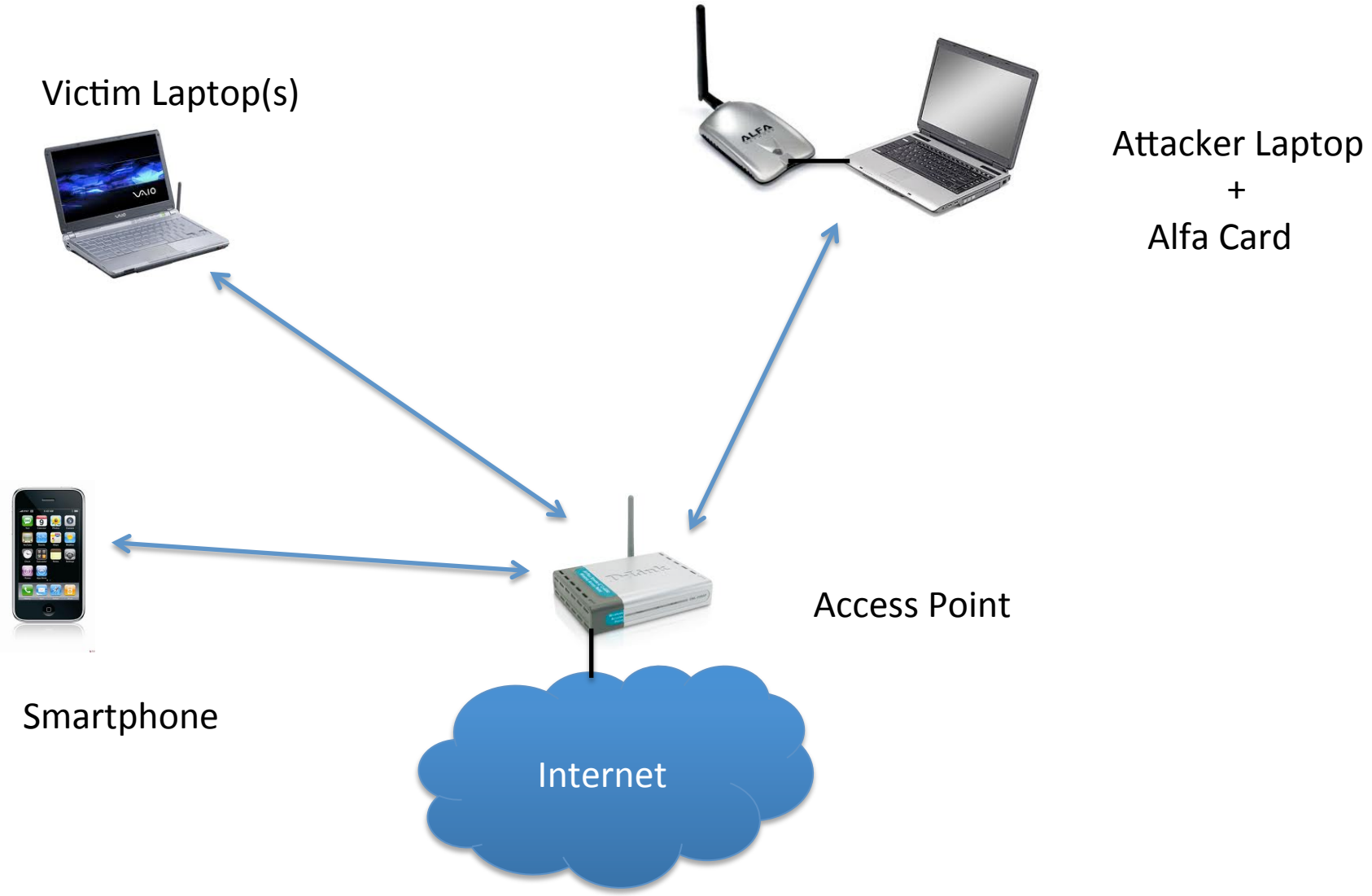
Difference between Wired and Wireless Sniffing

- Key difference with wired
  - Concept of channels and bands in wireless
  - **Wireless Card can only be on one channel at a time**
  - Cannot sniff on all channels and bands at the same time
  - **Wireless Card needs to be capable of operating in the given range : a?b?g?n?h?**
- Alfa Network card operates in b/g

Lab Session: Sniffing and Channel Hopping

- Use airodump-ng utility to cycle through the different channels
- Locate different wireless networks over the air
- View the packets in Wireshark

A Simple Wireless Network



Understanding WLAN Packets Types

- 3 types of packets:
  - Management
  - Control
  - Data
- Subtypes exist for each of the above
- Full details available in IEEE Specification

http://standards.ieee.org/about/get/802/802.11.html

Packet Sub-Types

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association request
00	Management	0001	Association response
00	Management	0010	Reassociation request
00	Management	0011	Reassociation response
00	Management	0100	Probe request
00	Management	0101	Probe response
00	Management	0110–0111	Reserved
00	Management	1000	Beacon
00	Management	1001	ATIM
00	Management	1010	Disassociation
00	Management	1011	Authentication
00	Management	1100	Deauthentication
00	Management	1101	Action
00	Management	1110–1111	Reserved
01	Control	0000–0111	Reserved
01	Control	1000	Block Ack Request (BlockAckReq)
01	Control	1001	Block Ack (BlockAck)
01	Control	1010	PS-Poll
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	ACK
01	Control	1110	CF-End
01	Control	1111	CF-End + CF-Ack

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
10	Data	0000	Data
10	Data	0001	Data + CF-Ack
10	Data	0010	Data + CF-Poll
10	Data	0011	Data + CF-Ack + CF-Poll
10	Data	0100	Null (no data)
10	Data	0101	CF-Ack (no data)
10	Data	0110	CF-Poll (no data)
10	Data	0111	CF-Ack + CF-Poll (no data)
10	Data	1000	QoS Data
10	Data	1001	QoS Data + CF-Ack
10	Data	1010	QoS Data + CF-Poll
10	Data	1011	QoS Data + CF-Ack + CF-Poll
10	Data	1100	QoS Null (no data)
10	Data	1101	Reserved
10	Data	1110	QoS CF-Poll (no data)
10	Data	1111	QoS CF-Ack + CF-Poll (no data)
11	Reserved	0000–1111	Reserved

Source: IEEE 802.11-2007 Standard

Understanding the Access Point

- Access Point is configured with an SSID
- This SSID acts as a network name for discovery
- Clients search for this access point or network using this SSID
- Access Point sends out broadcast frames called Beacon Frames to announce its presence
- Clients use this to show available wireless networks list

Demo

- Start Wireshark and capture Beacon Frames
- Analyze various important header fields in the Beacon Frame
- Identify things like SSID, Encryption, Channel etc.

TaDa! Pwning Beacon Frames

- Anyone can create and transmit beacon frames
- All clients will list that as a new access point
- We will use MDK on BT4 to do this

Demo Time!

What did we learn?

- Spoofing 802.11 frames is simple
- No protection mechanism available
- Seems similar to TCP/IP spoofing
- We will use this “insecurity” over and over again in attacks

Objective

- To understand how AP and Clients communicate
- Strip down to the packet level
- Understand details with Wireshark

Demo Time!

- Create an open authentication and no encryption based AP with SSID “SecurityTube”
- Connect a client to it
  - Laptop
  - Smartphone
- Collect all the packets using Wireshark
  - Ensure your card is also on the same channel
- Analyze the flow

Client – AP Connection Packets

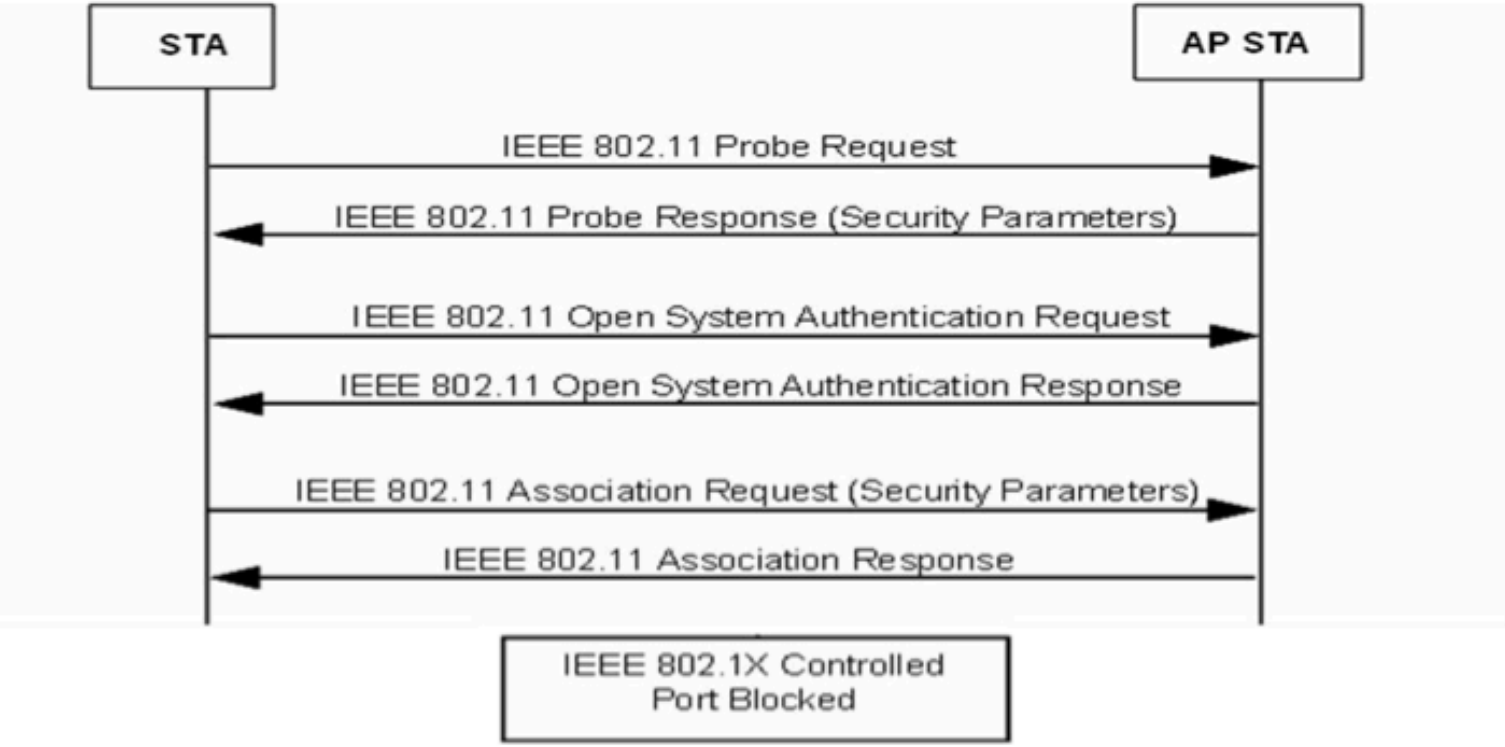
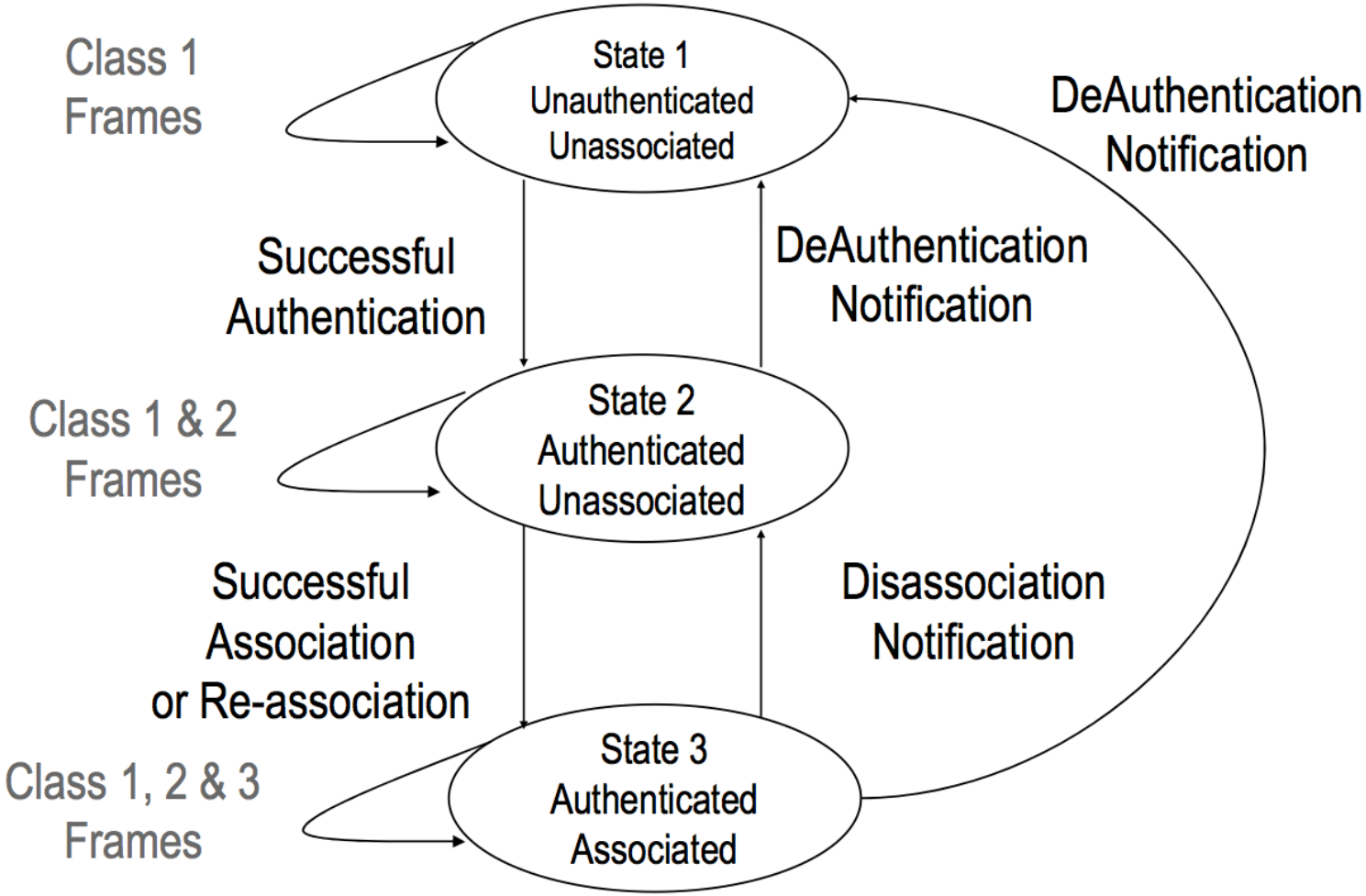


Figure 5-11—Establishing the IEEE 802.11 association

Source: IEEE Standard

AP-Client State Machine

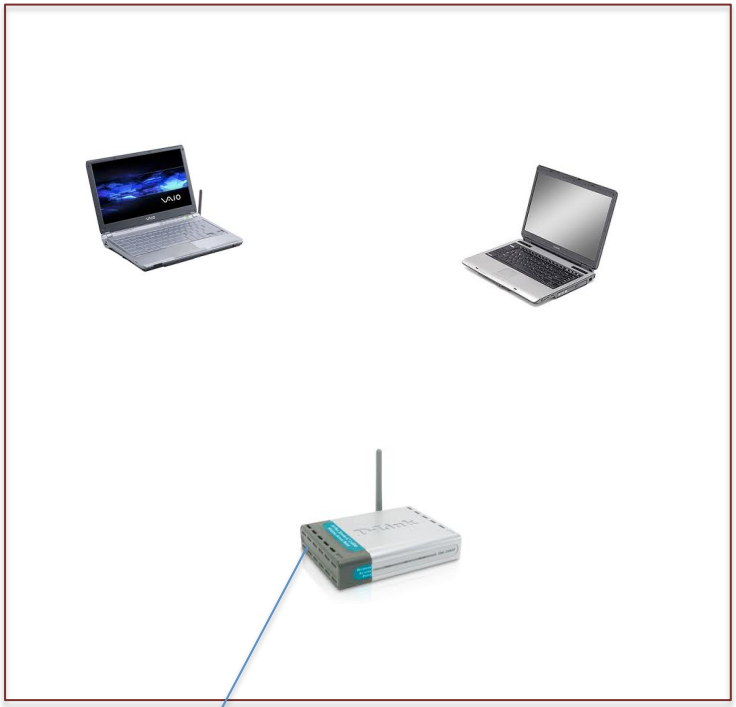


Terminology

- STA – STAtion (Wireless Client)
- BSS – Basic Service Set (set of nodes communicating with each other)
  - Infrastructure BSS (AP and Clients)
  - Independent BSS (Ad-Hoc Clients)
- ESS – Extended Service Set (set of connected BSSs)
- BSSID – Basic Service Set Identifier
  - Infrastructure BSS (MAC address of AP)
  - IBSS (Randomly Chosen MAC address)
- DS – Distribution System (connects APs in an ESS)

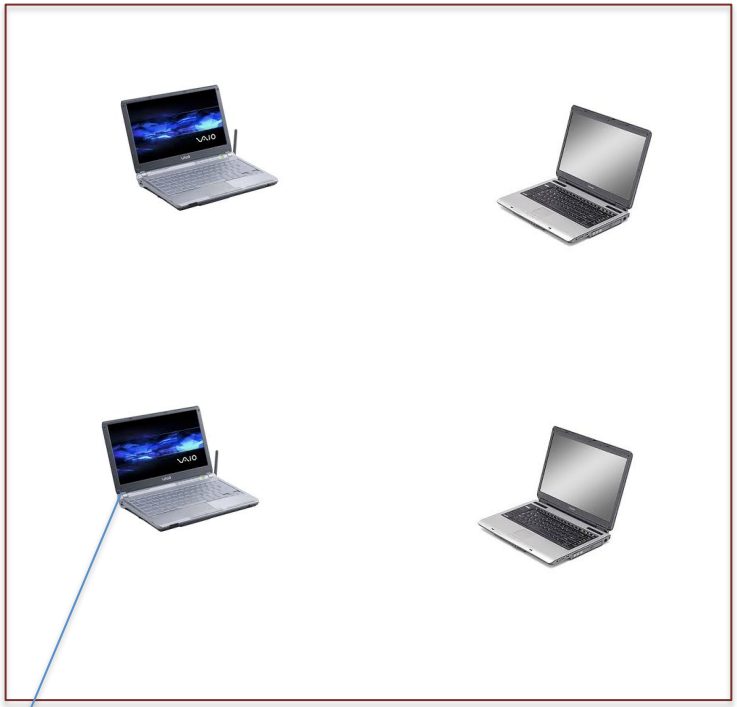
BSS

Infrastructure BSS



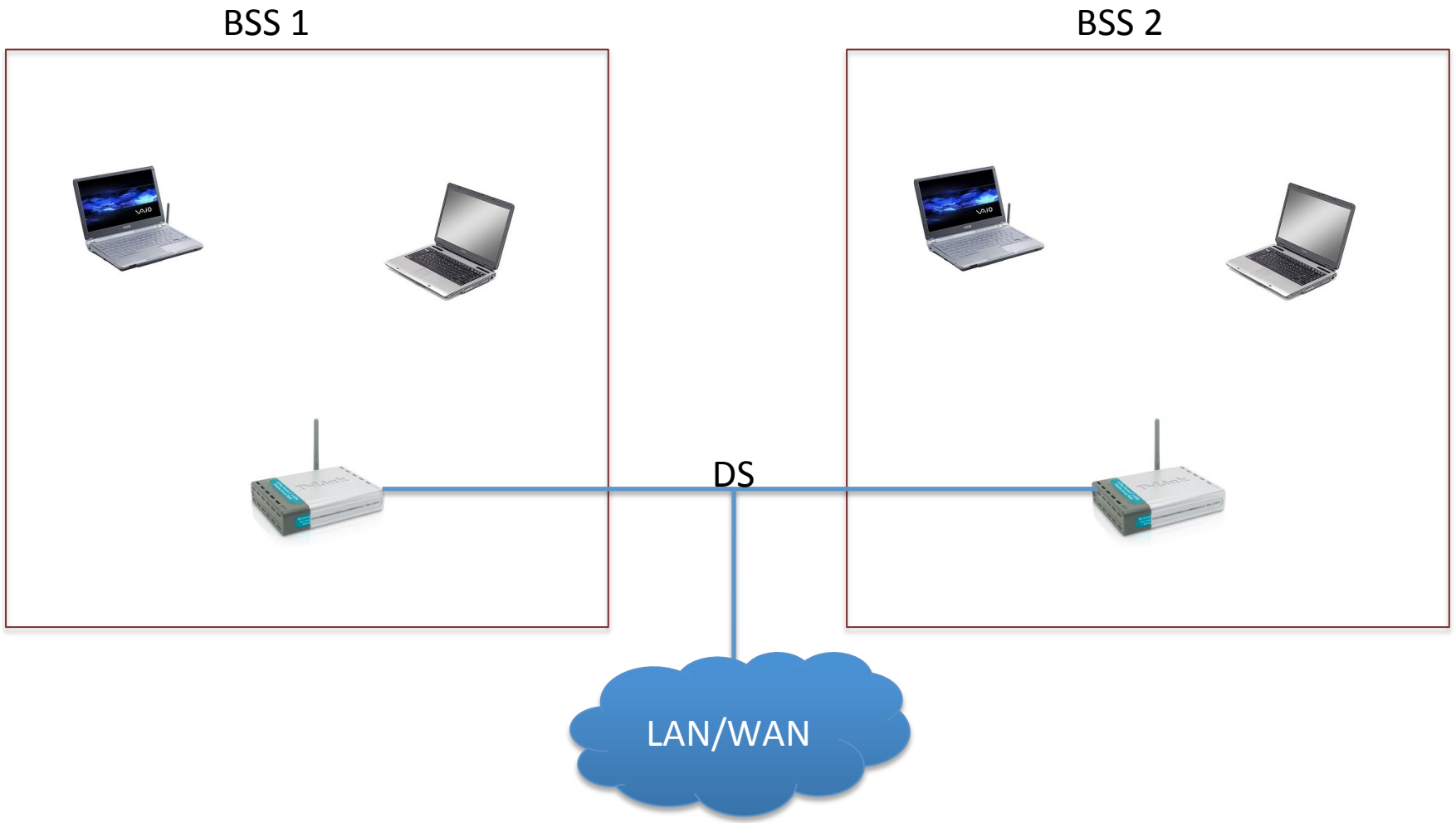
BSSID = MAC of AP

Independent BSS (Ad-Hoc)

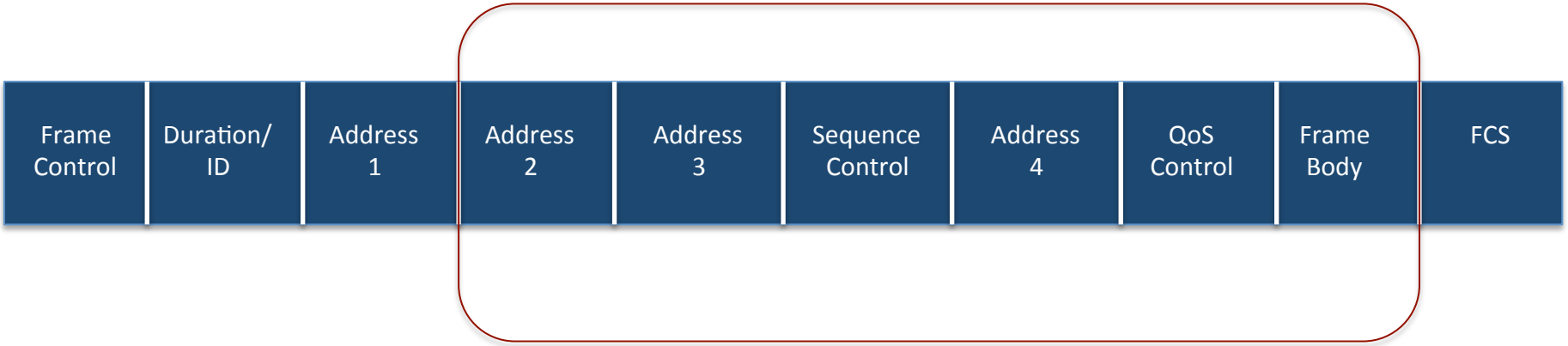


BSSID = Random MAC chosen by First Client in Ad-Hoc Mode

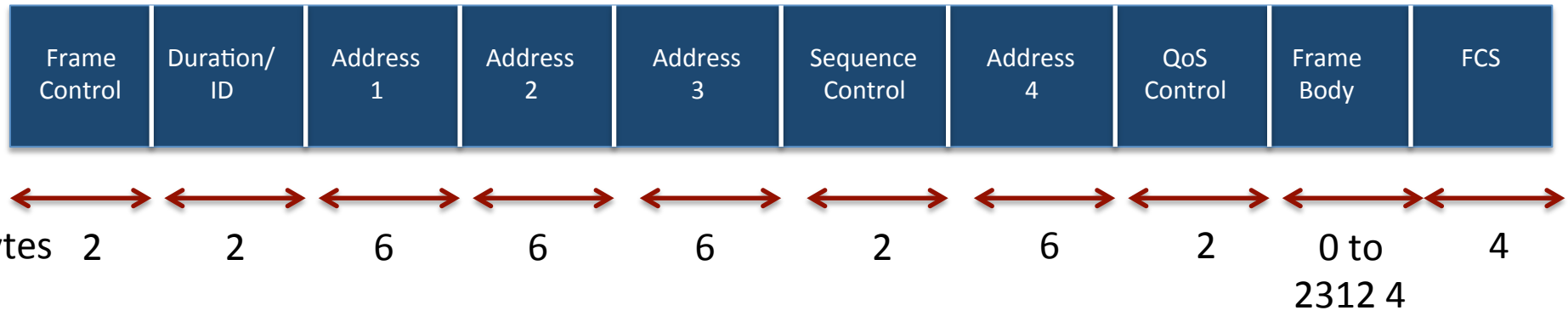
ESS



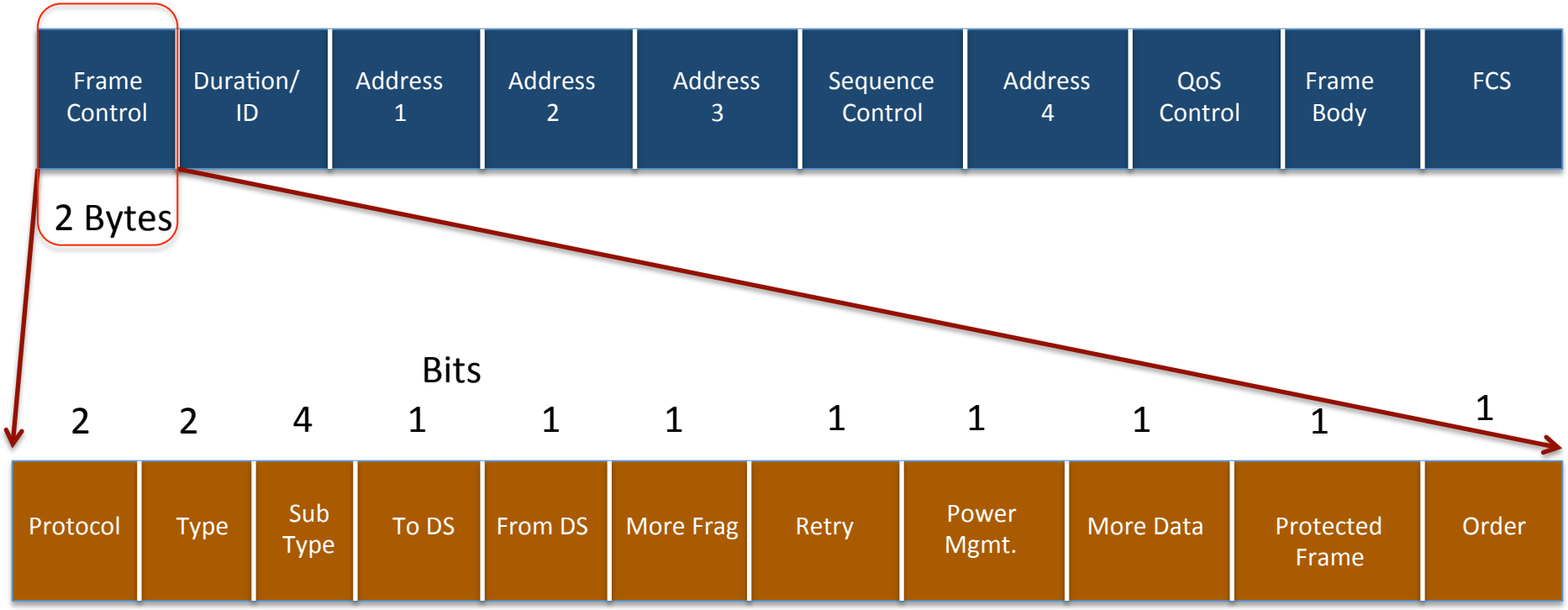
WLAN Packet Header



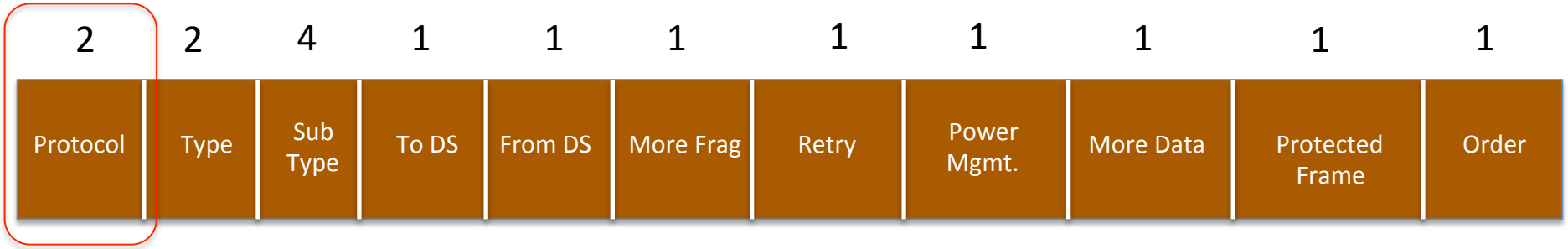
Presence Depends on Packet Type / Sub Type



Frame Control Field

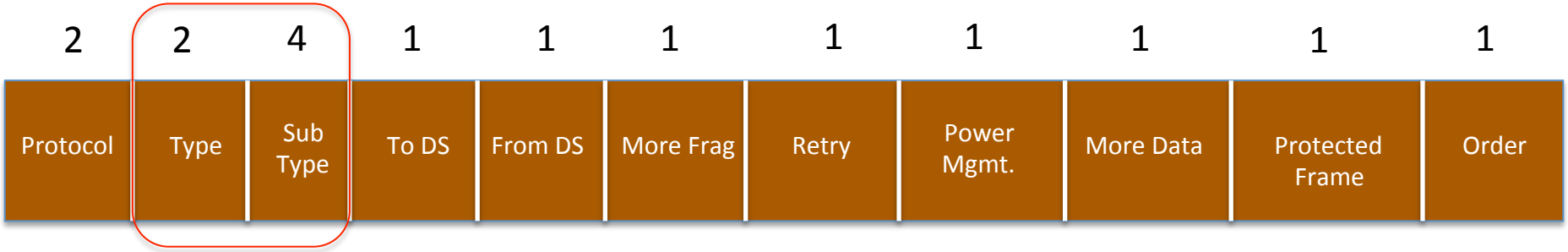


Protocol



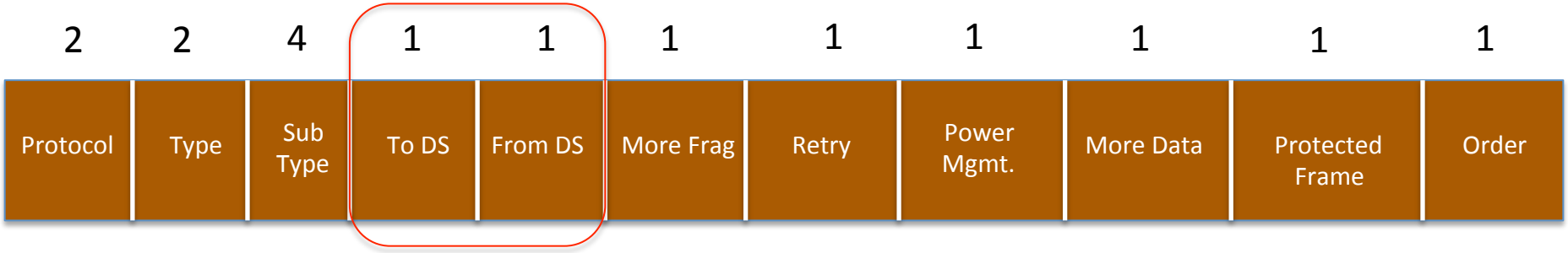
- Default to “0” value
- May change when a major revision happens incompatible with the previous version

Type and SubType



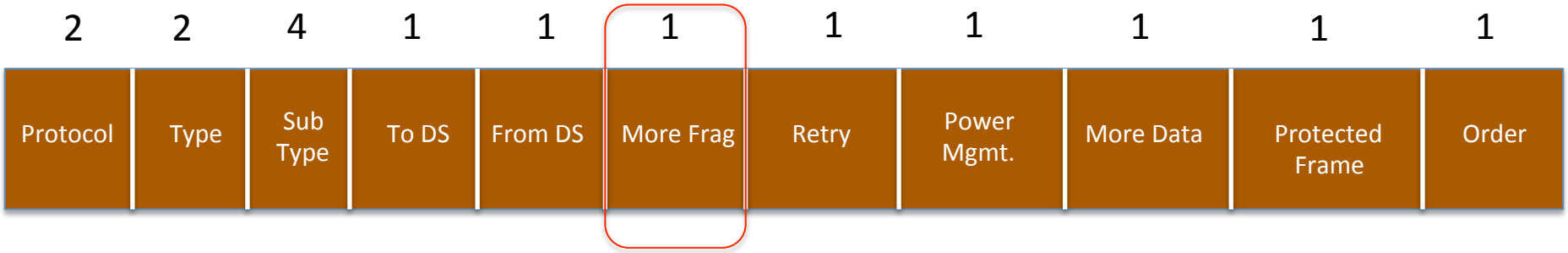
- Type – Management, Control and Data Frames
- Sub-Types in each

To and From DS



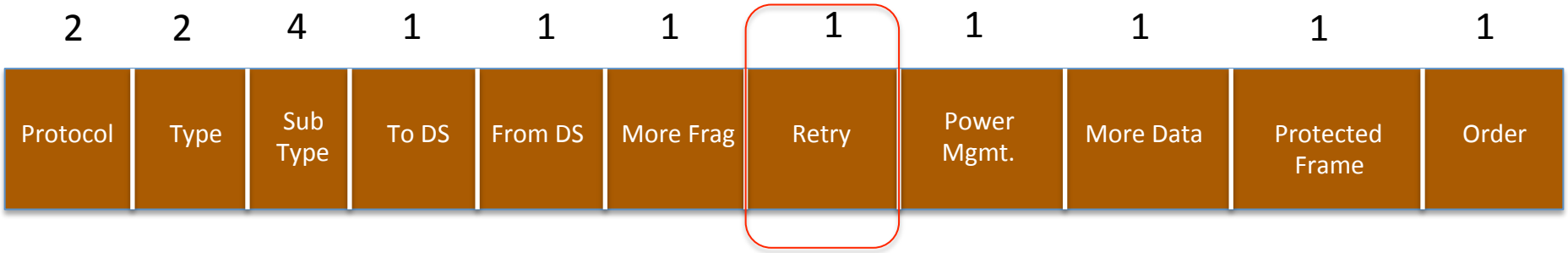
To DS	From DS	Interpretation
0	0	STA to STA in same IBSS (Ad-Hoc), Management and Control Frames
0	1	Exiting the Distribution System (DS)
1	0	Entering the DS
1	1	Used in Wireless Distribution Systems (WDS)

More Frag



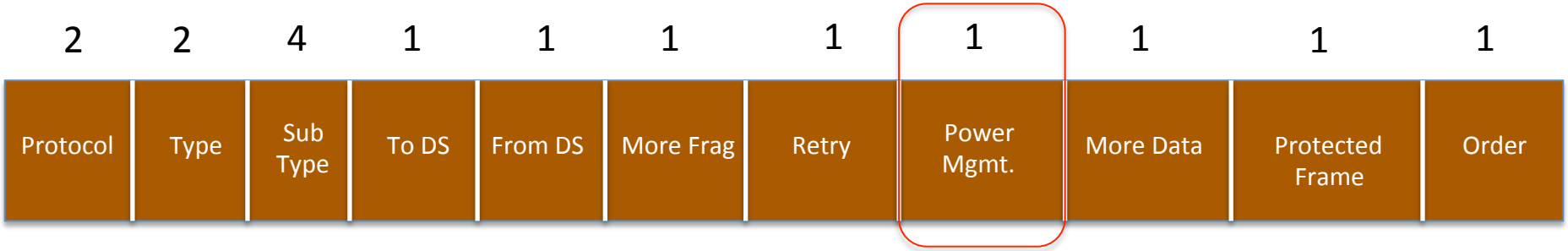
- Indicates if more fragments of the current frame are to follow
- Only application to Data and Management frames

Retry



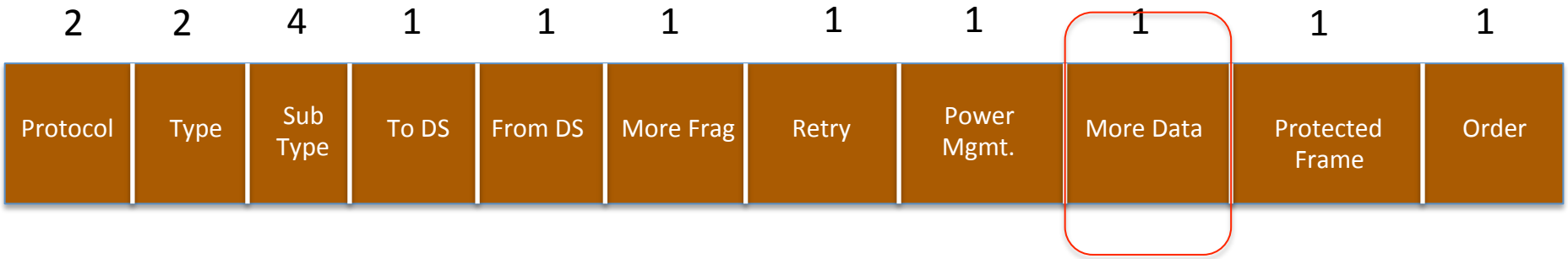
- Indicates is current frame is a retransmission
- Applicable to Management and Data Frames only

Power Management



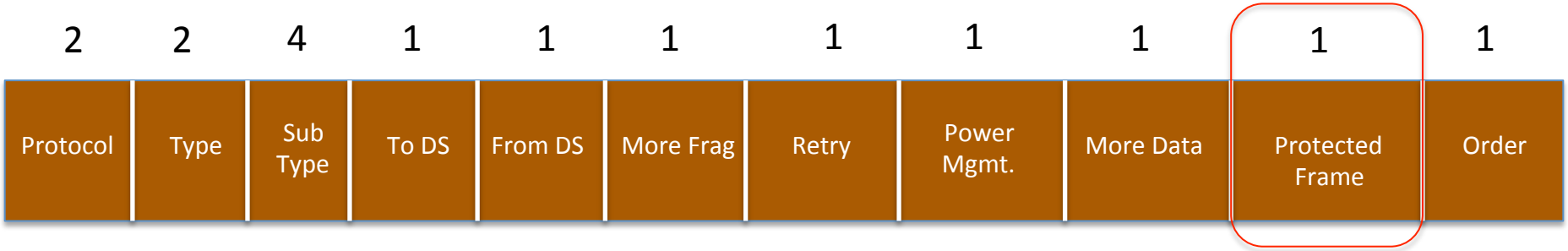
- Indicates if the STA is in Power Save Mode or Active Mode

More Data



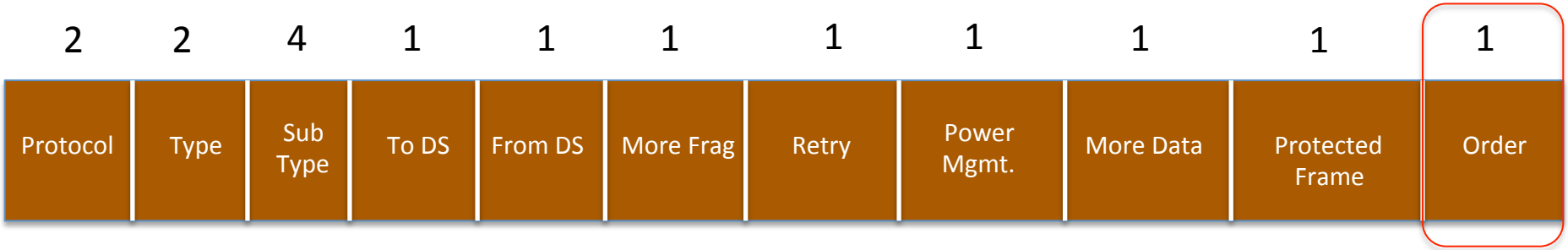
- Indicates to an STA in Power Save mode that more data is to follow
- Data is queued up on the AP

Protected Frame



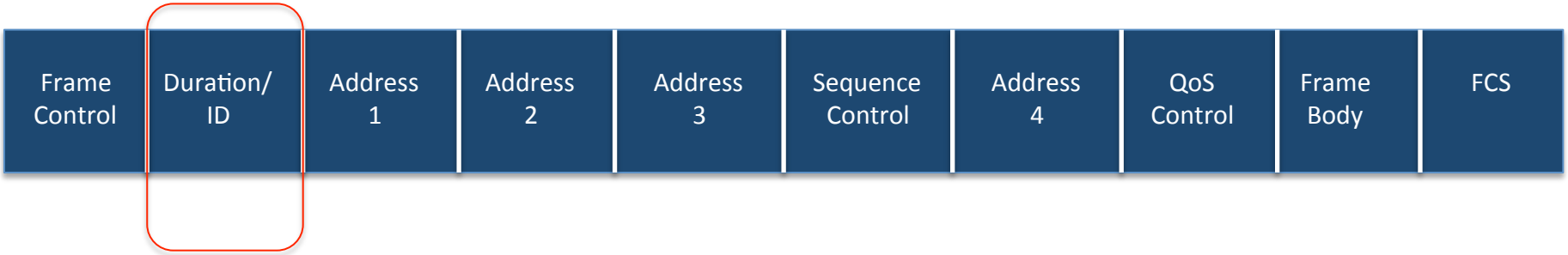
- 1 indicates that the Frame Body is encrypted
  - Data frames
  - Management frames of Type Auth
- 0 indicates no encryption

Order



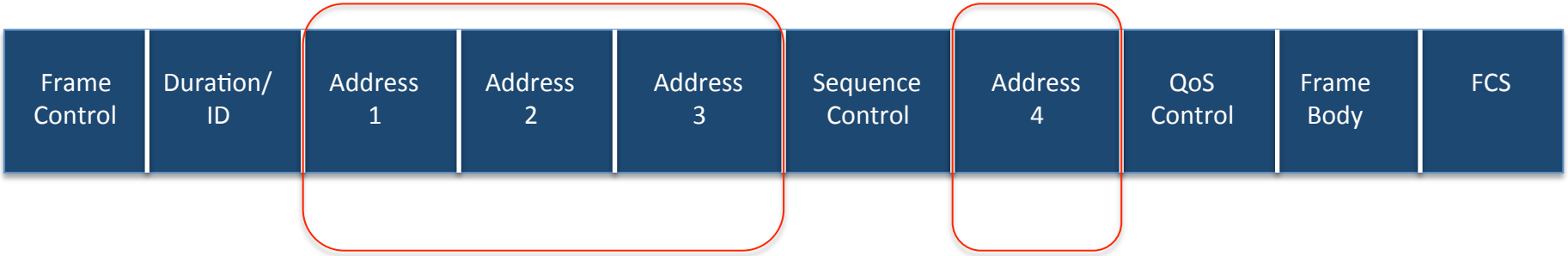
- Indicates that all received frames must be processed in order

Duration



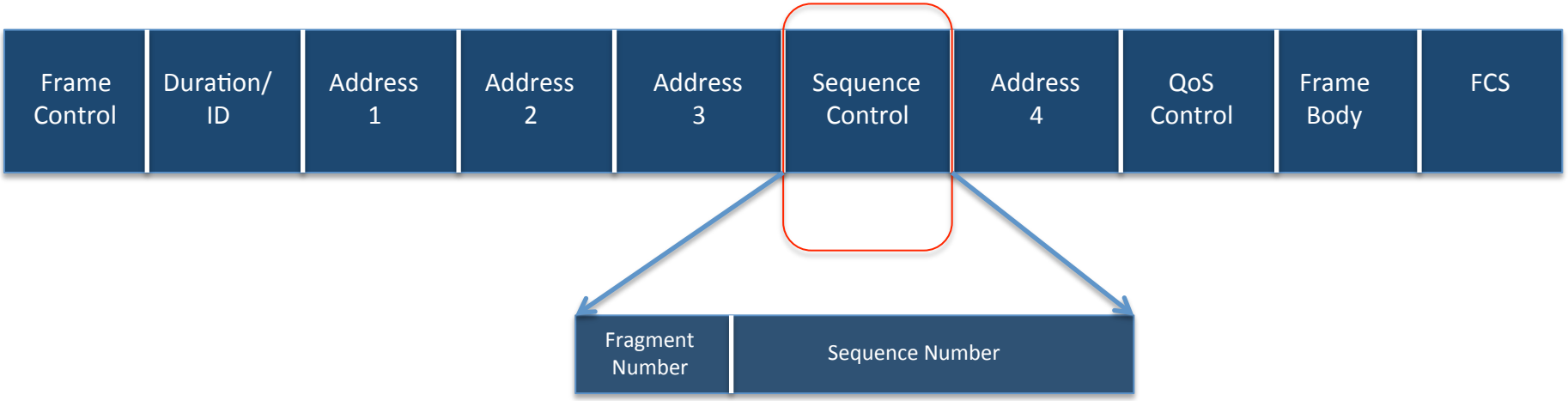
- Used to set the Network Allocation Vector (NAV) ☺
- NAV is the minimum amount of time a STA needs to wait before attempting transmission
- Also used in CFP and PS-Poll frames

Address



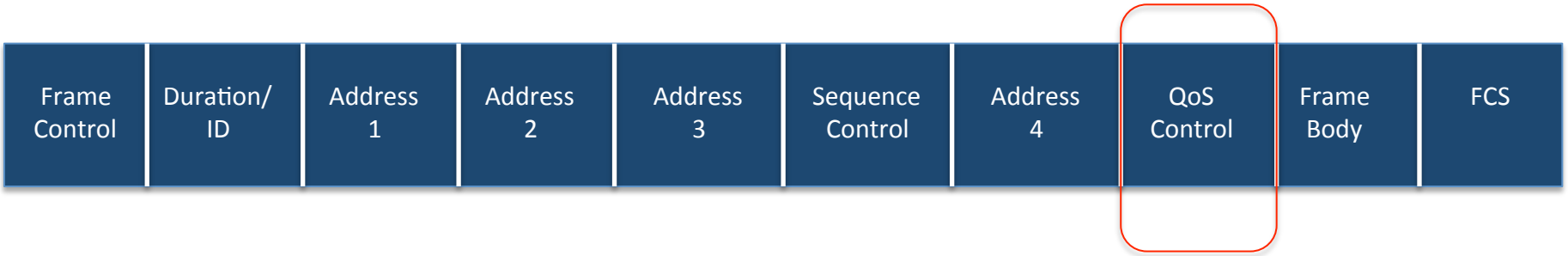
- Value and Presence depends on Type/Sub-Type
- Destination Address
- Source Address
- BSSID

Sequence Control



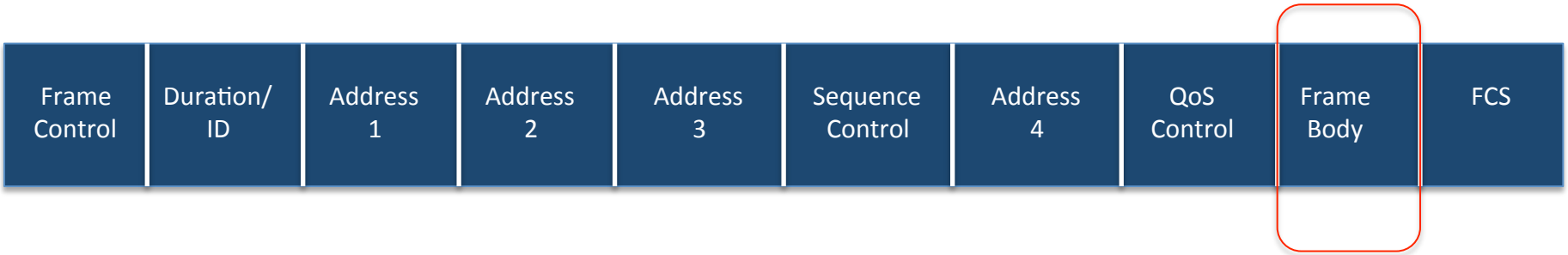
- Sequence number of the packet
- Fragment number of the packet

QoS Control



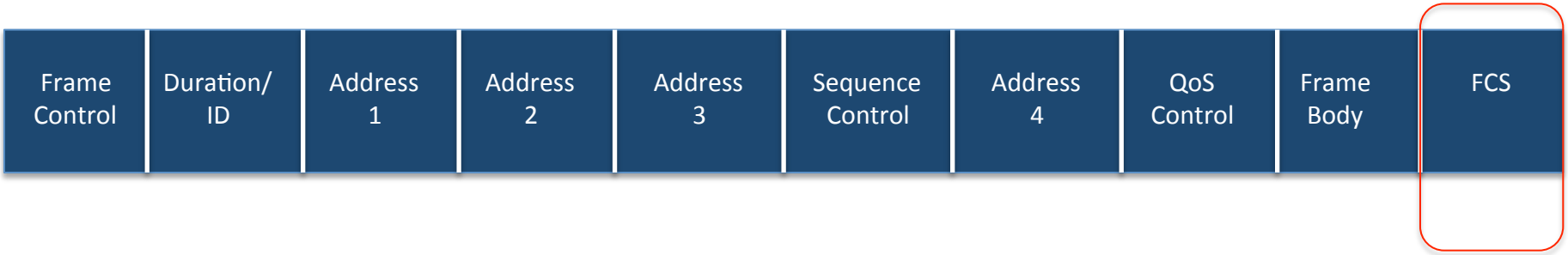
- Quality of Service Related
- In Data Frames

Frame Body



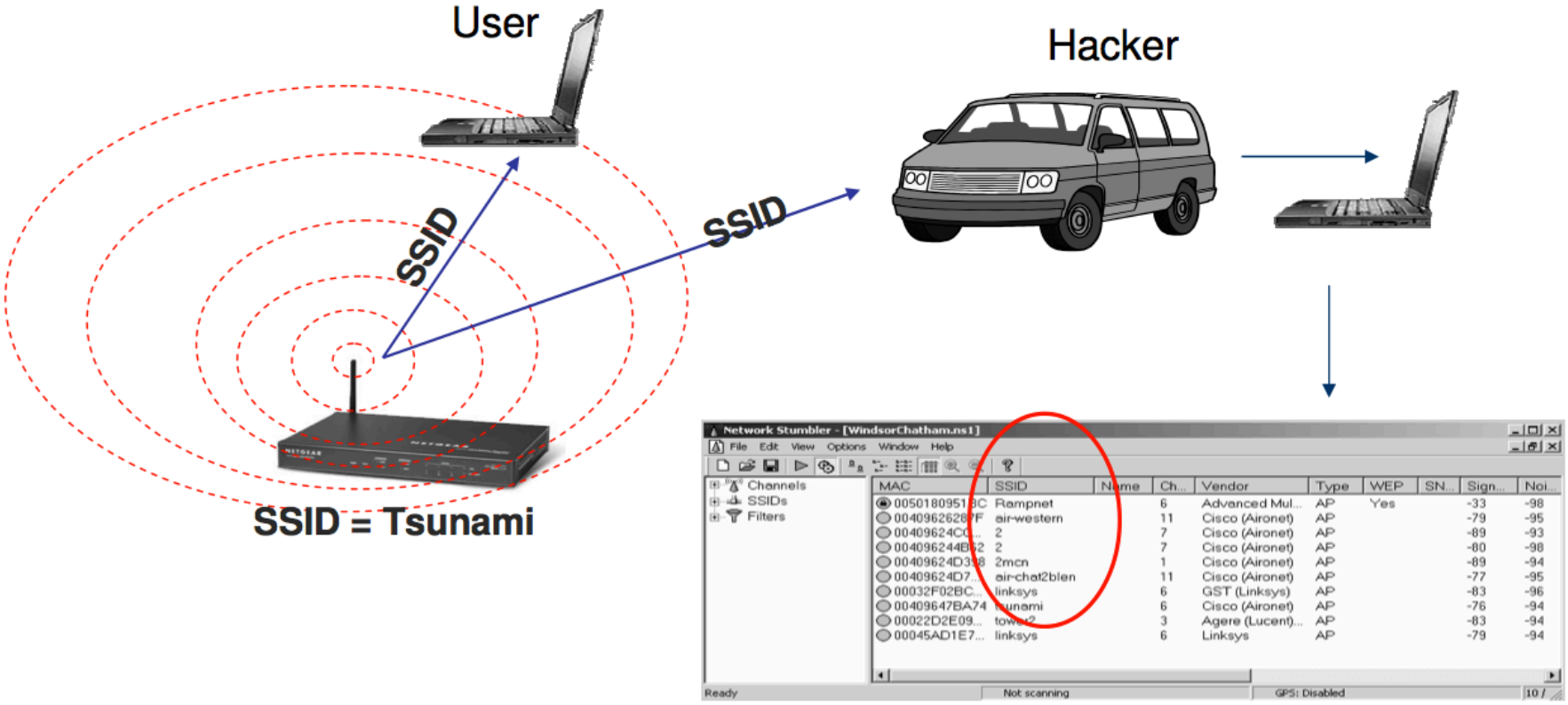
- Contains the data payload
  - Management frame details
  - Actual data

FCS



- CRC check over the MAC header and Frame Body
- Easy to beat ☺

Sniffing SSIDs



Hidden SSID

- Turn SSID Broadcasting off in Beacon Frames
- Just monitoring Beacon Frames will not give you the SSID
- A “Security through Obscurity” technique at best
- Can only deter novices
- Hardly a challenge for the experienced wireless hacker

Pwning Hidden SSIDs

- Multiple Techniques:
  - Monitor Air for a new Client trying to associate with the access point (passive)
  - De-authenticate one or all clients and monitor reconnections (active)
- Basic idea is to force the network to send Probe / Association packets
- These packets contain the SSID even if not present in the Beacon frame from the access point

Origin of MAC Filters

- Used in the Wired World
- Switches and Filtering devices like Firewalls
- Idea was to have a set of “whitelisted” MAC addresses and deny rest
- Is insecure as MAC address can be easily spoofed
- Reasonably secure if authorized MAC addresses are few and attacker cannot get physical access to the authorized machines to find the MAC

Wireless MAC Filters

- Not a feature in the 802.11 standard
- Can add them on the access point (network layer filter)
- Simple way to only allowed whitelisted MACs
- ** Time to Laugh ** ☺ ☺ ☺
  - MAC addresses are visible in plain text in the WLAN header
  - We simply need to monitor associated clients and find their MAC addresses
  - Use the MAC when the Client is gone / still present
  - No defense at all!

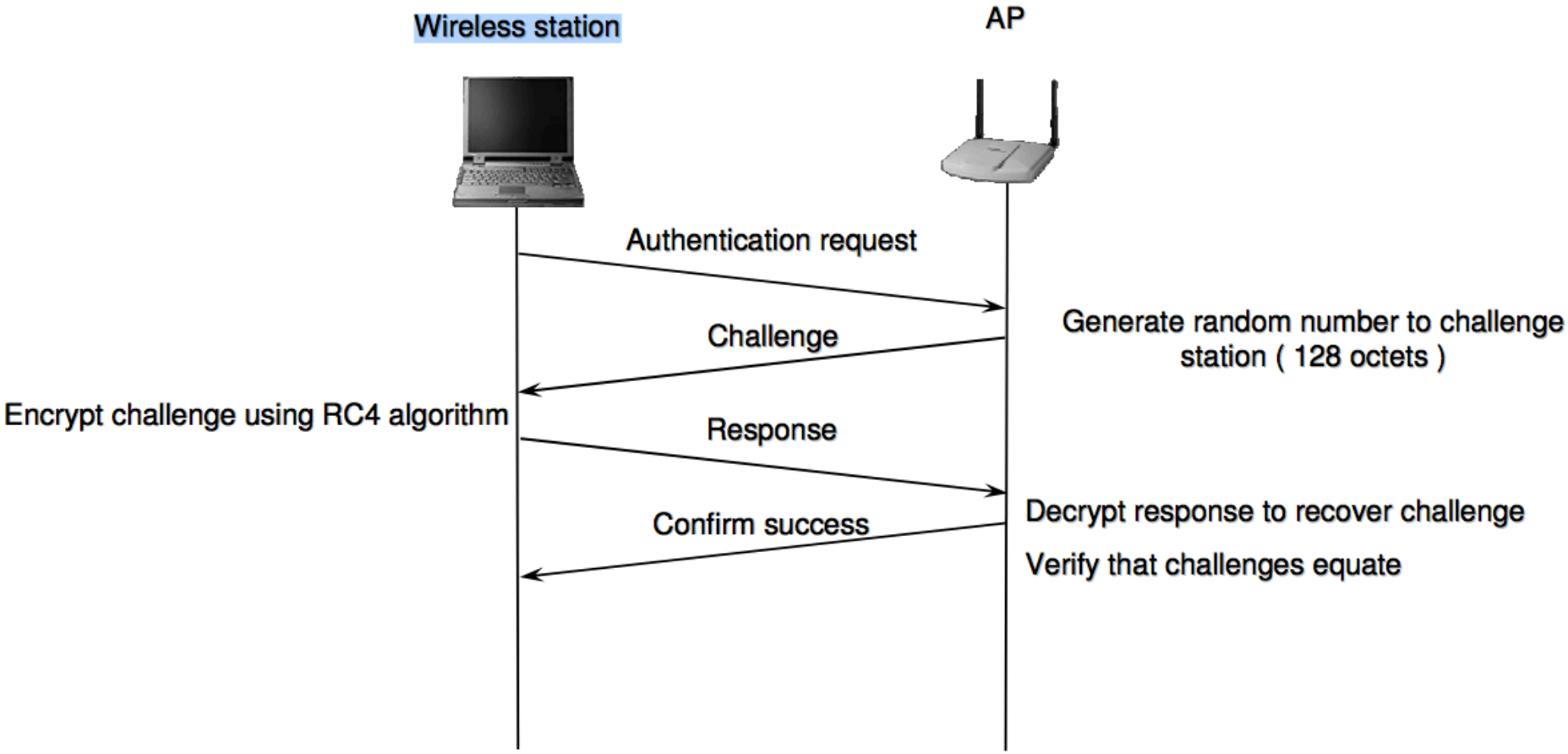
WLAN Authentication

- WLAN Authentication by itself is not powerful at all
- 2 types:
  - Open Authentication
  - Shared Authentication

Open Authentication

- No “actual” Authentication mechanism at all
- 2 packets exchanged between Client and AP, and authentication ends
- Cases where authentication may fail
  - MAC Filtering

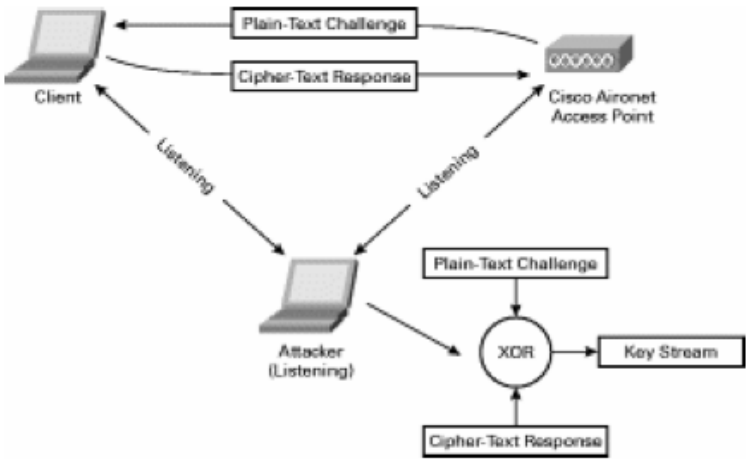
Shared Authentication



Understanding Shared Authentication

- Challenge is encrypted using the WEP key
- WEP uses RC4 which is a stream cipher
- RC4 Keystream is XOR'ed with Plain Text challenge and response is returned
- We will discuss WEP in detail later

Simple Math to nail Shared Auth



X – Plain Text Challenge
Y – WEP Keystream
Z – Encrypted Challenge

Z = X (xor) Y

Z (xor) X = (X (xor) Y) (xor) X = Y

Using the Keystream and IV

- Use for shared authentication with the AP
- Can be used to encrypt small packets (128 bytes)
  - Arbitrary injection
- IV and Keystream can be harvested to create a table based decryption attack
  - Need a lot of SKA tries
  - Can only decrypt first 128 bytes of every packet

Demo Time

- Setup AP to use WEP and Shared Key Auth
- Try connecting without knowing the key
- Sniff the packets and dump the keystream
- Use this to pwn shared authentication

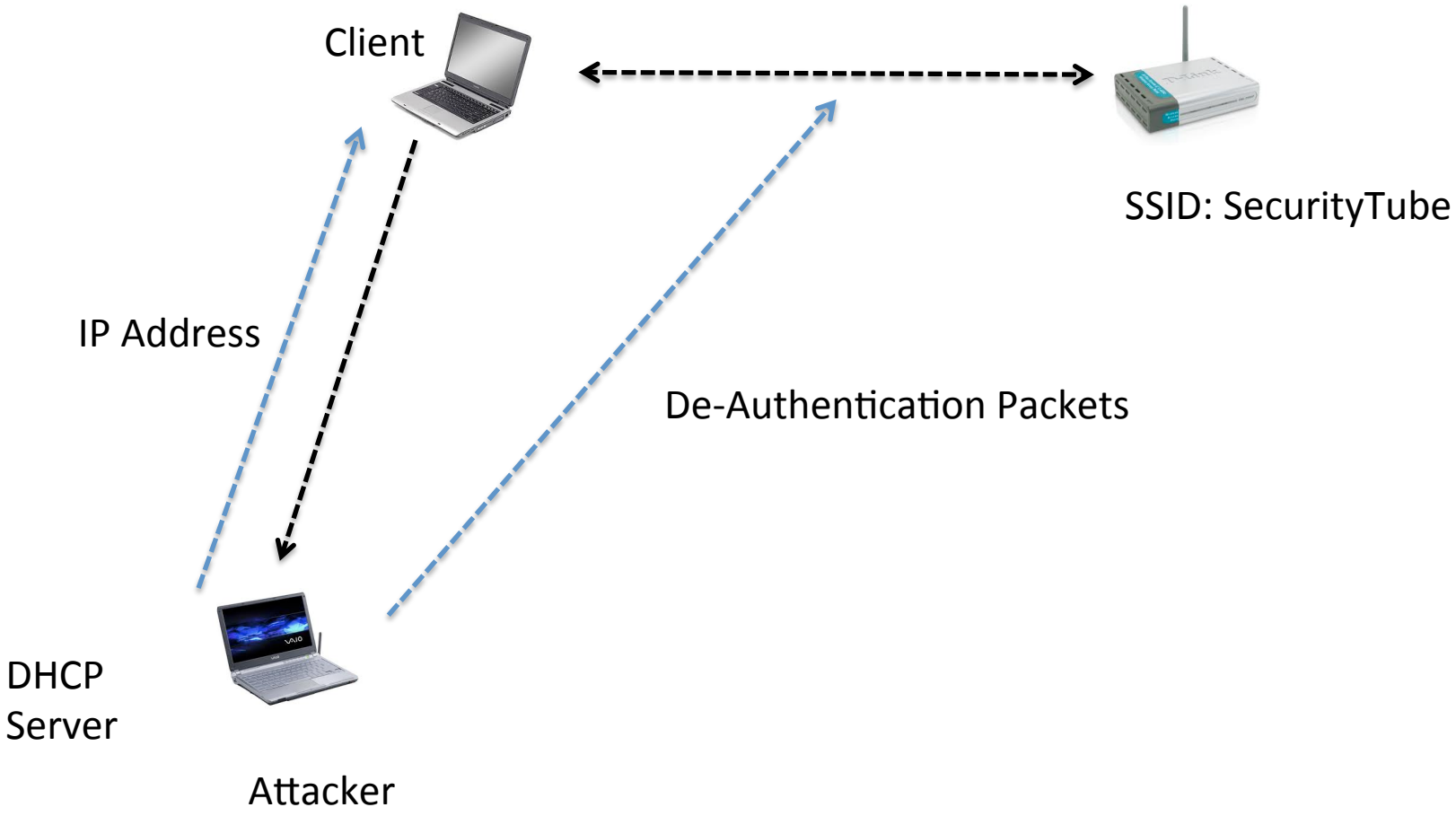
Hotspot Basics

- Free / Paid WiFi based internet offered in public places
  - Coffee shops
  - Airport
- Typically uses
  - Open Authentication
  - MAC Filtering at times
  - No Encryption
    - Distribution of keys would be a nightmare
  - Can use captive portals for application layer authentication

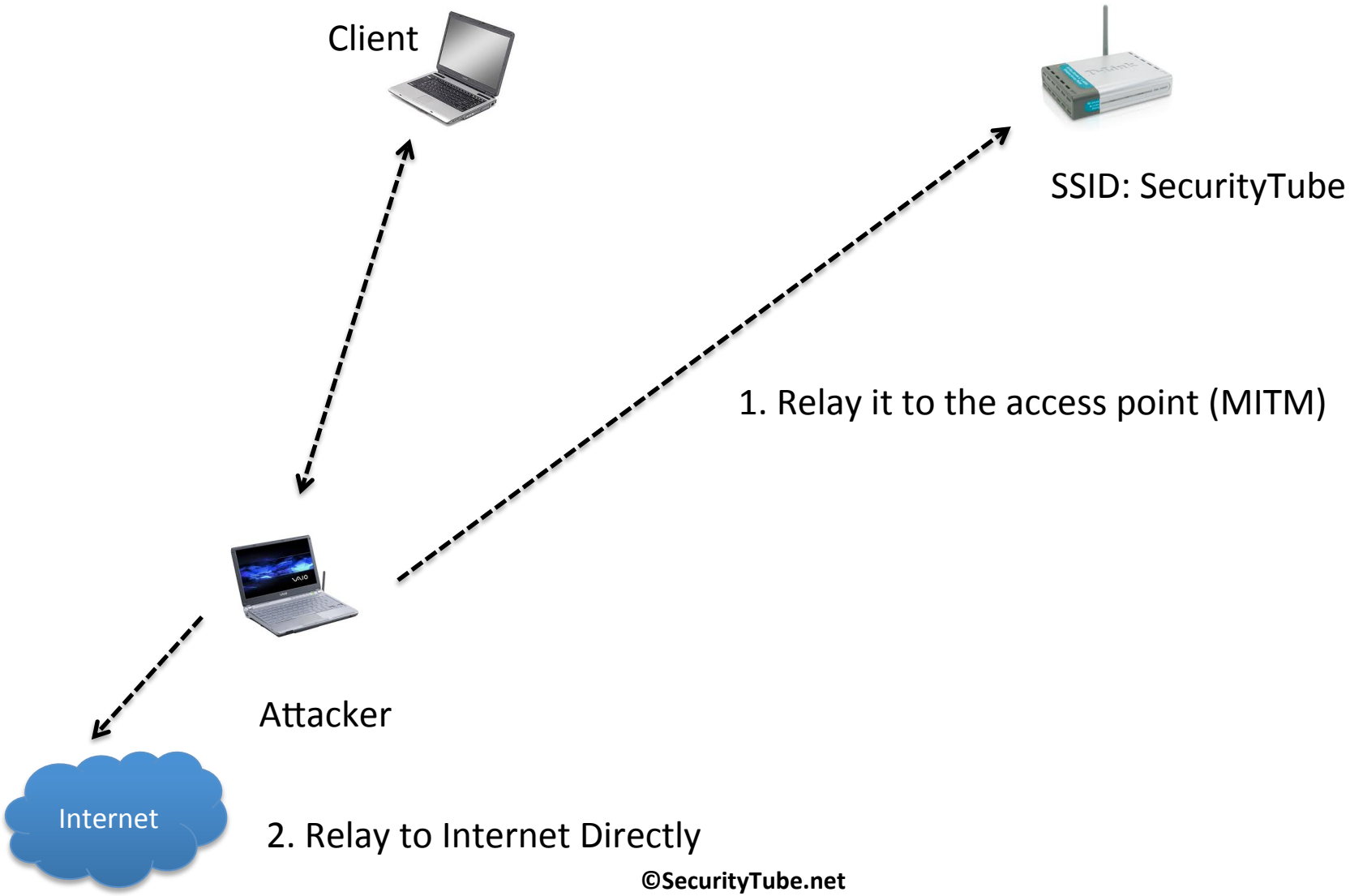
Hotspot Attacks

- Create an Evil Twin in the vicinity
  - Same ESSID
  - Same BSSID (optional)
- Use De-Authentication attacks to break Client AP Connection
- If Evil Twin has higher signal strength, then Client will connect to it

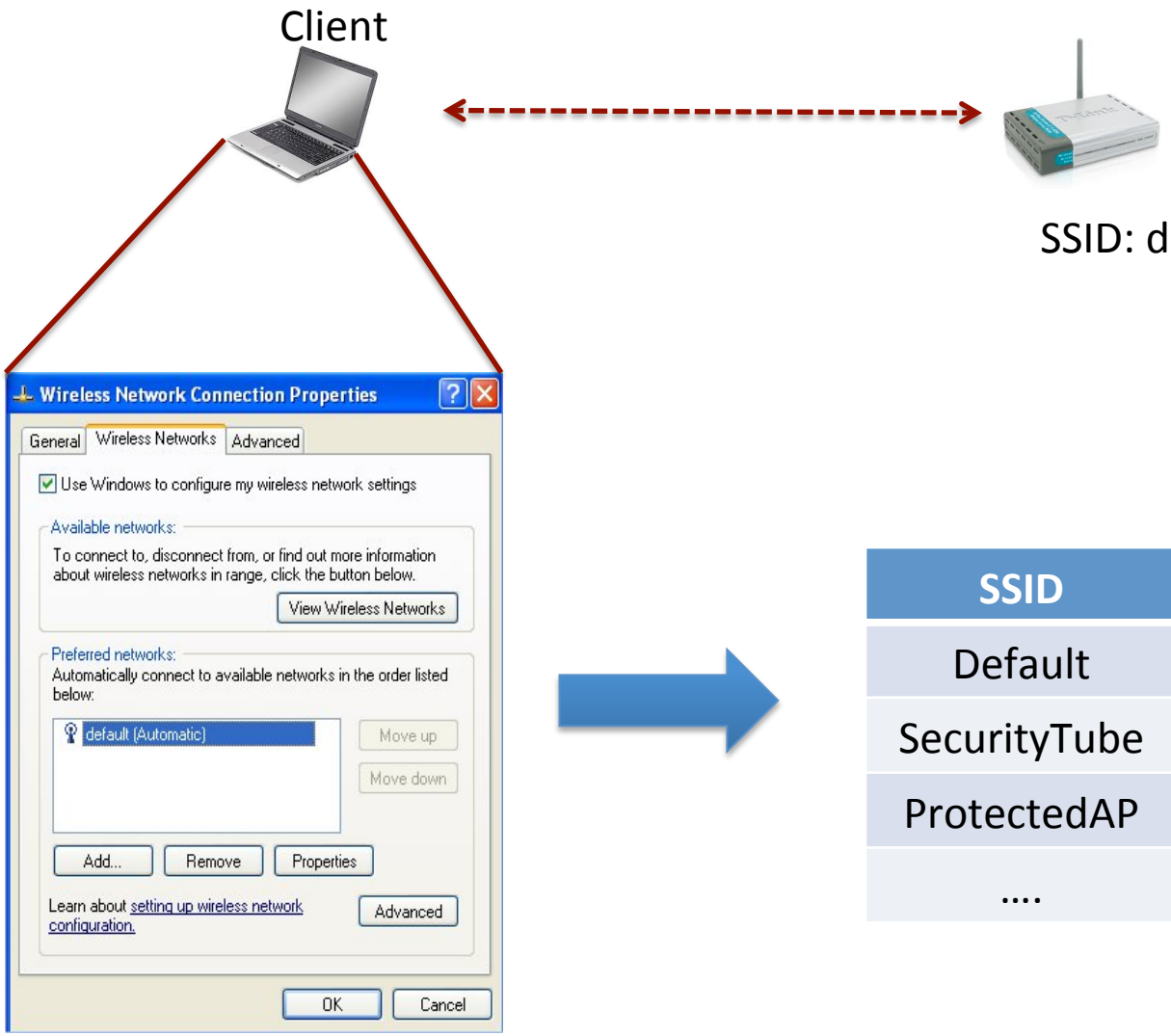
Attack Visualization



Post Attack Options

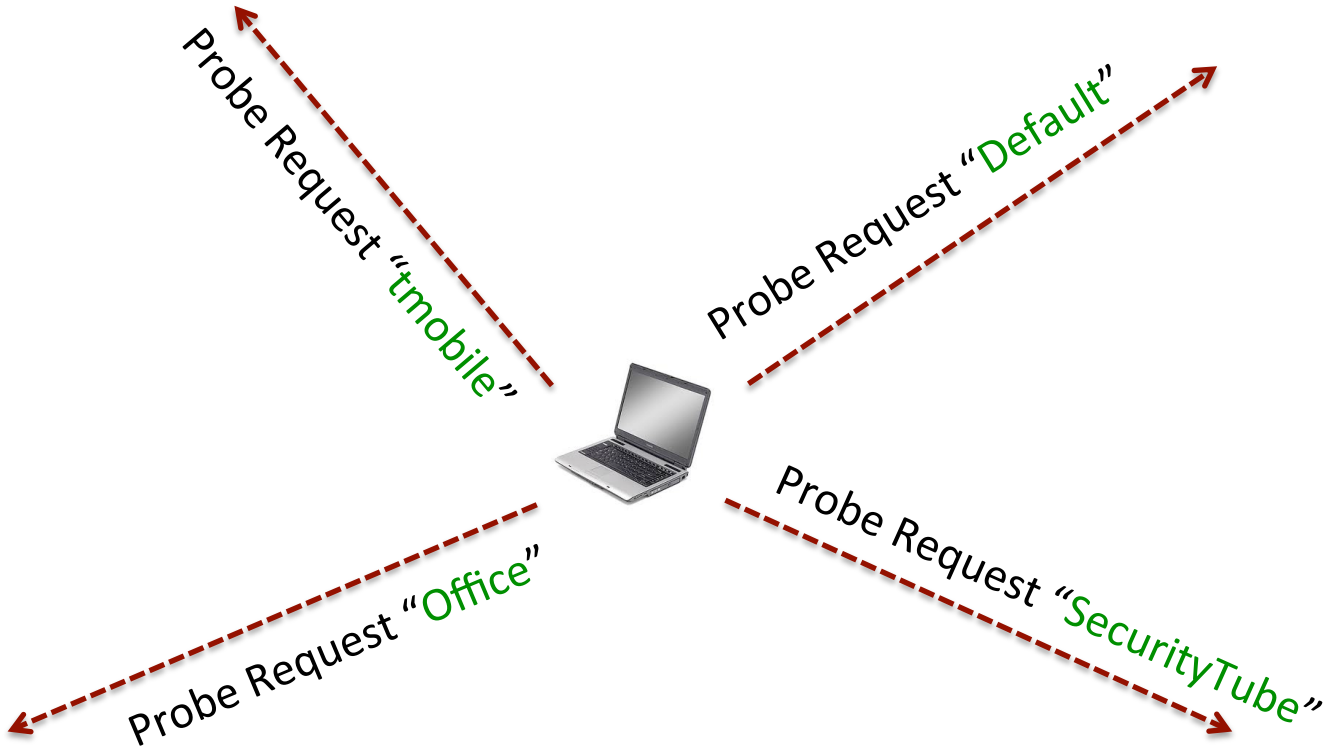


Understanding Clients



SSID	Credentials
Default	...
SecurityTube	...
ProtectedAP	********
....	...

An Isolated Client



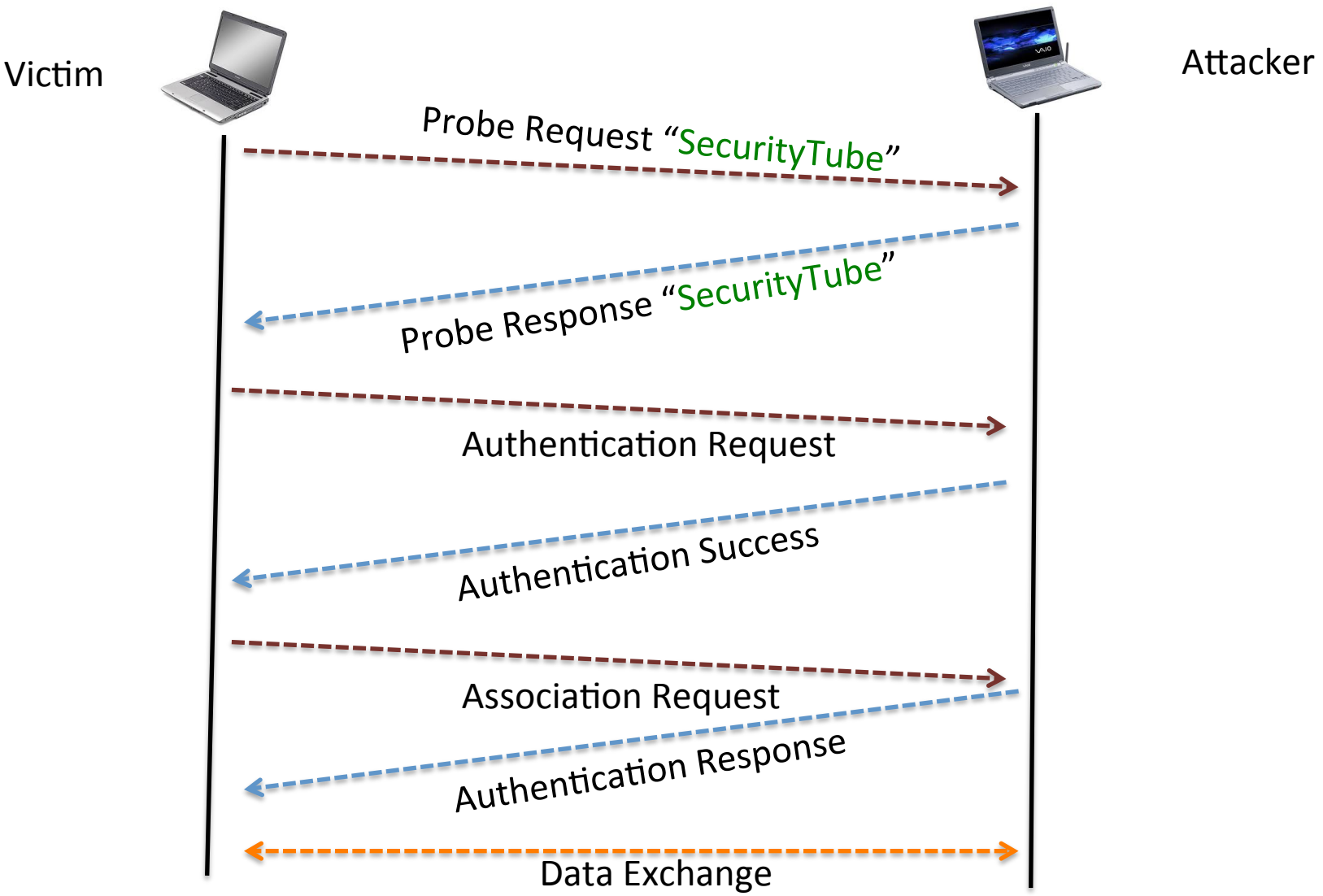
Inconsistent Behavior

- Different OSs behave differently
  - Linux
  - Windows
  - OS X
- Difference in Behavior even between SP in windows
- We will take up most common behavior
  - Client searching for known access points

Multiple Cases Possible

- Access Point stored in the PNL or similar could have either of 3 configurations:
  - No Encryption
  - WEP
  - WPA/WPA2
- We will deal with each of the them separately

Case 1: Open Authentication, No Encryption



Fundamental Problem

- Client cannot authenticate the access point
- The SSID all alone is used to decide whom to connect to
- Anyone can set a similar SSID and force a client to connect to their access point
- This is especially true with Hotspot SSIDs as they by definition are Open Authentication with no Encryption

Case 2 and Case 3

- WEP and WPA/WPA2
- Shared Key Authentication

- We will talk about these once we finish the encryption fundamentals class

Operating Frequency Range and Regulations

Table 3-2 Operating Frequency Range for 802.11b and 802.11g

Channel Identifier	Center Frequency	FCC (America)	ESTI (EMEA)	TELEC (Japan)	MOC (Israel Outdoor)[1]
1	2412	X	X	X	
2	2417	X	X	X	
3	2422	X	X	X	
4	2427	X	X	X	
5	2432	X	X	X	X
6	2437	X	X	X	X
7	2442	X	X	X	X
8	2447	X	X	X	X
9	2452	X	X	X	X
10	2457	X	X	X	X
11	2462	X	X	X	X
12	2467		X	X	X
13	2472		X	X	X
14[2]	2484			X	

[1] Israel allows channels 1 through 13 indoors.

[2] Japan requires a special license for channel 14.

Understanding Transmit Power

dBm to Watt Conversion Chart

dBm = log10 (mW)*10
mW =10^(dBm/10

40 dBm	10.00 watts	
36 dBm	4.00 watts	< Maximum ERP allowed by FCC in U.S.
30 dBm	1.00 watts	
27 dBm	500 milliwatts	
26 dBm	400 milliwatts	
25 dBm	320 milliwatts	
24 dBm	250 milliwatts	
23 dBm	200 milliwatts	
22 dBm	160 milliwatts	
21 dBm	130 milliwatts	
20 dBm	100 milliwatts	< Maximum ERP allowed by E.T.S.I. In Europe
15 dBm	32 milliwatts	
10 dBm	10 milliwatts	
5 dBm	3.2 milliwatts	
4 dBm	2.5 milliwatts	
3 dBm	2.0 milliwatts	
2 dBm	1.6 milliwatts	
1 dBm	1.3 milliwatts	
0 dBm	1.0 milliwatts	
1- dBm	0.79 milliwatts	
5- dBm	0.32 milliwatts	
10- dBm	0.1 milliwatts	
20- dBm	0.01 milliwatts	
30- dBm	0.001 milliwatts	
40- dBm	0.0001 milliwatts	
50- dBm	0.00001 milliwatts	
60- dBm	0.000001 milliwatts	
70- dBm	0.0000001 milliwatts	
80- dBm	0.00000001 milliwatts	< Receive threshold for most WLAN devices

EIRP

Effective Isotropic Radiated Power (EIRP)

= Transmitter Power (in dBm)

+ Antenna Gain (in dBi)

- Cable loss (in dBm)

Can this setting be changed?

- Yes ☺ We can change our channel (without any driver or kernel modifications) to any one of the following:

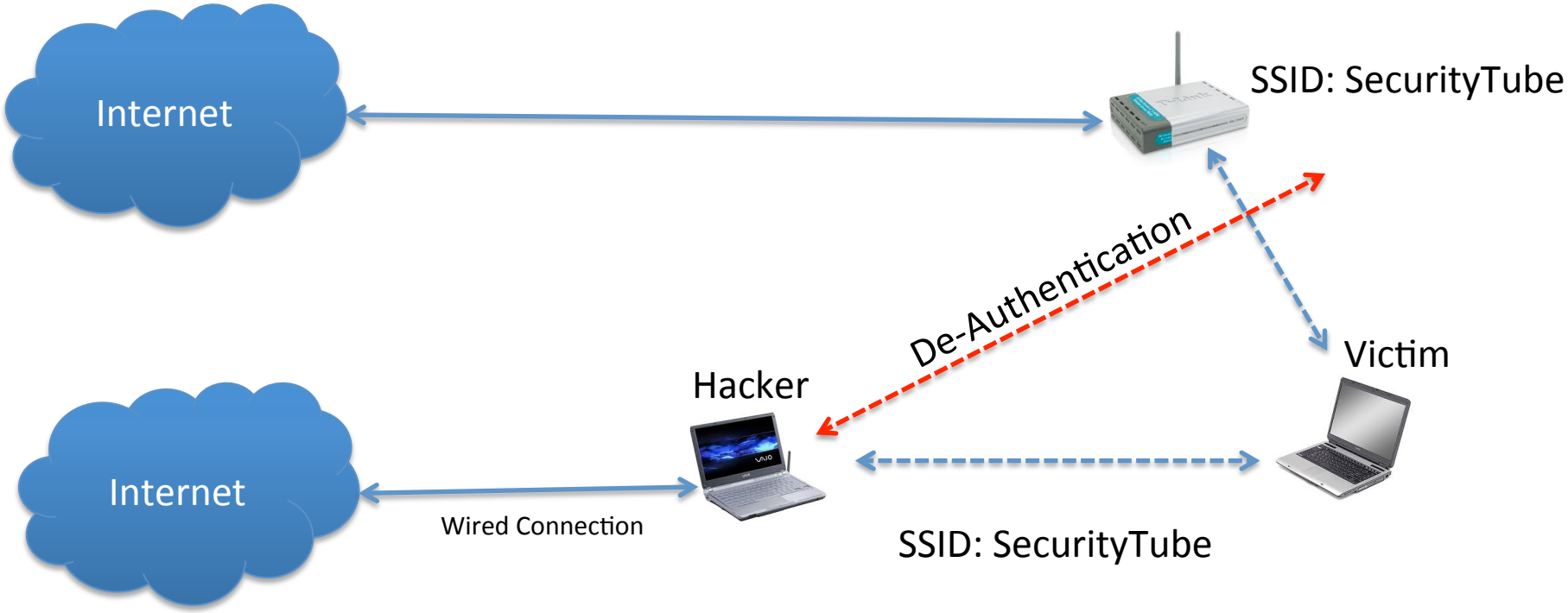
http://git.kernel.org/?p=linux/kernel/git/linville/wireless-regdb.git;a=blob;f=db.txt;hb=HEAD

- To be used when you are traveling to a new country
- The card will need to support the channel and max transmit power for the country
- Might be illegal to transmit high power or use other channels in your country

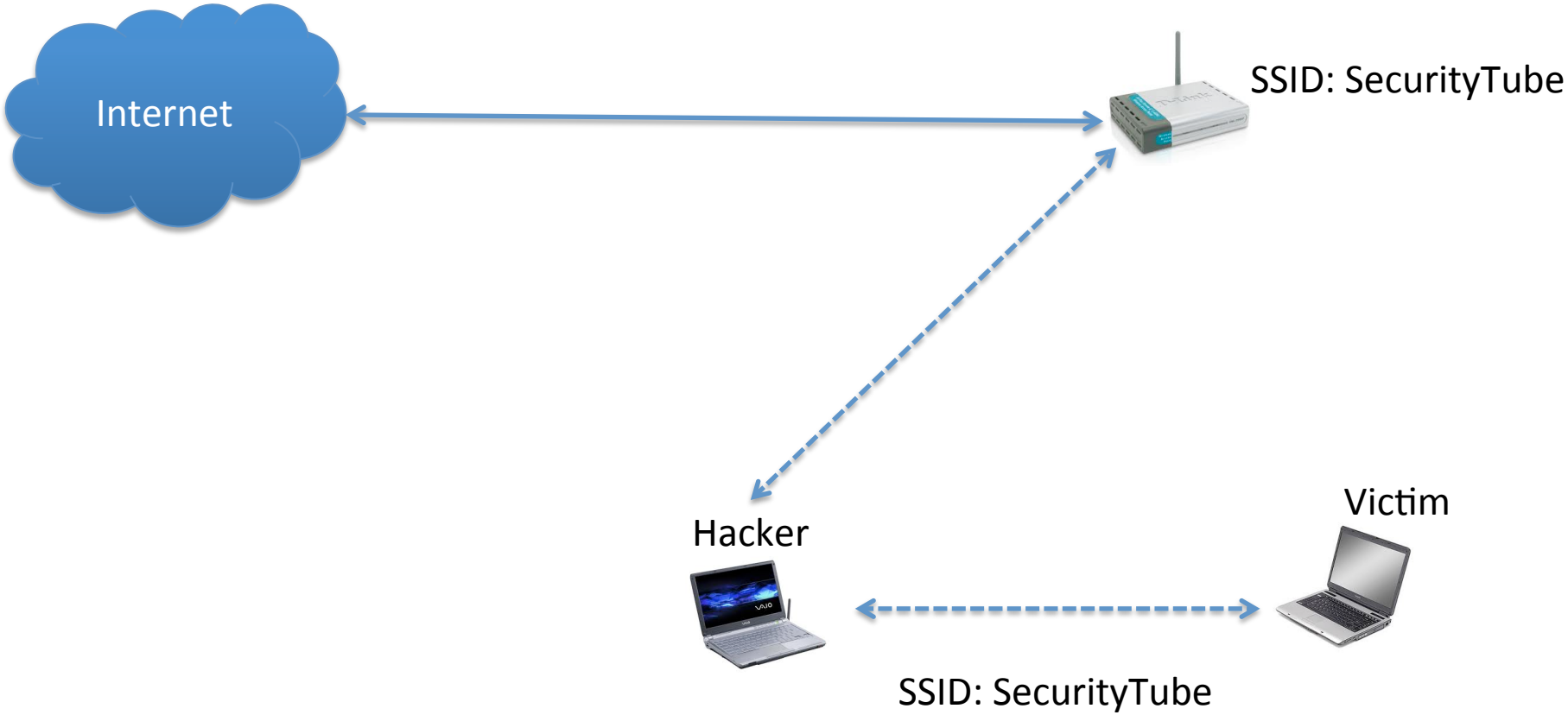
Bolivia and Belize to Alfa's Rescue

```
108
109 country BO:
110         (2402 - 2482 @ 40), (N/A, 30)
111         (5735 - 5835 @ 40), (N/A, 30)
112
113 country BR:
114         (2402 - 2482 @ 40), (N/A, 20)
115         (5170 - 5250 @ 40), (3, 17)
116         (5250 - 5330 @ 40), (3, 20), DFS
117         (5490 - 5710 @ 40), (3, 20), DFS
118         (5735 - 5835 @ 40), (3, 30)
119
120 country BY:
121         (2402 - 2482 @ 40), (N/A, 20)
122         (5170 - 5250 @ 40), (N/A, 20)
123         (5250 - 5330 @ 40), (N/A, 20), DFS
124         (5490 - 5710 @ 40), (N/A, 27), DFS
125
126 country BZ:
127         (2402 - 2482 @ 40), (N/A, 30)
128         (5735 - 5835 @ 40), (N/A, 30)
```

Wireless MITM



Variation 1



Variation 2

SSID: SecurityTube

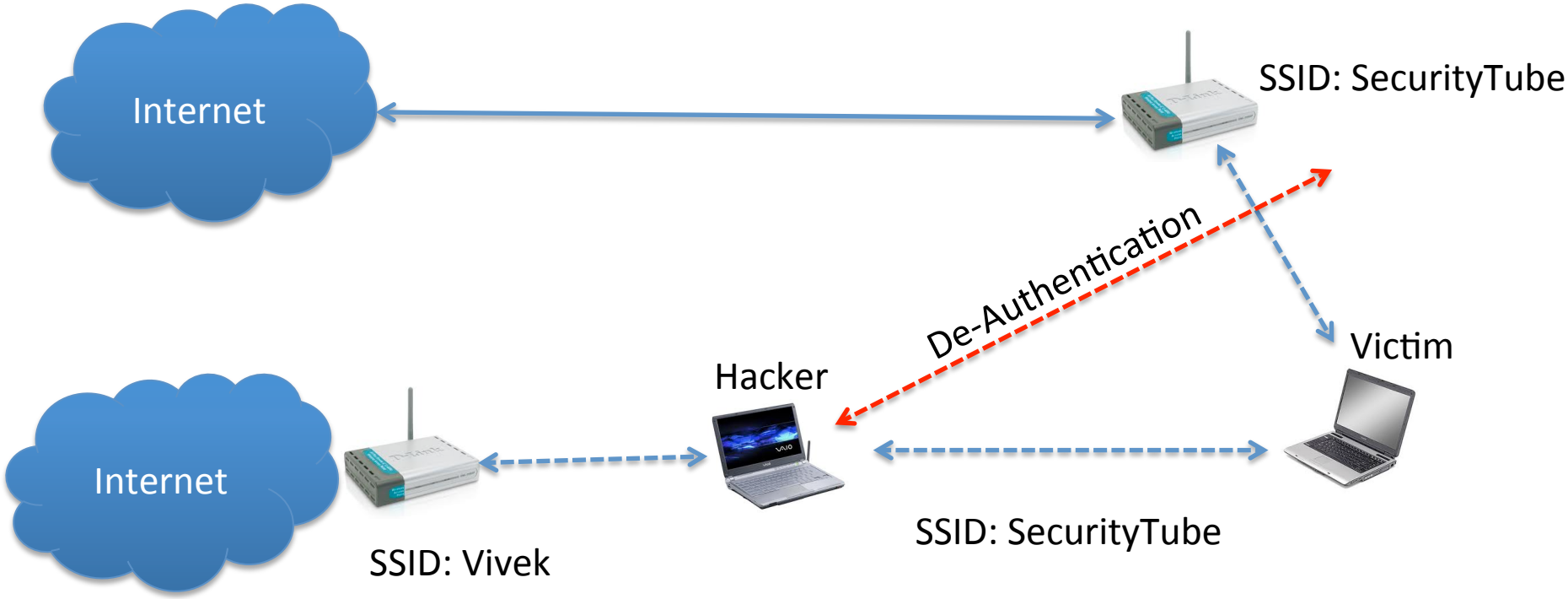
GPRS
3G
4G

Hacker

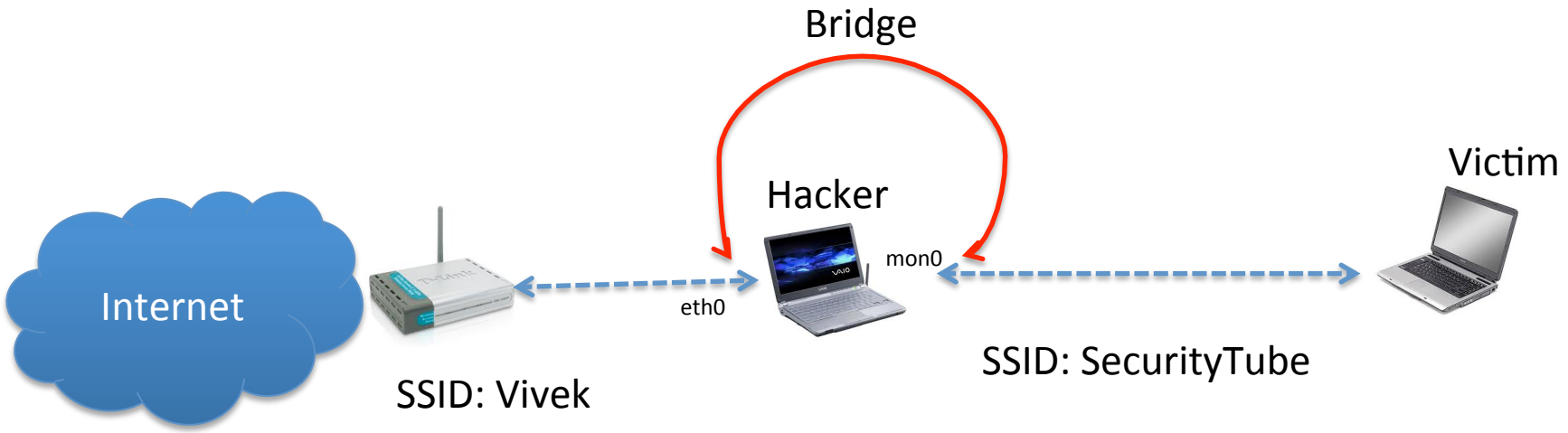
Victim

SSID: SecurityTube

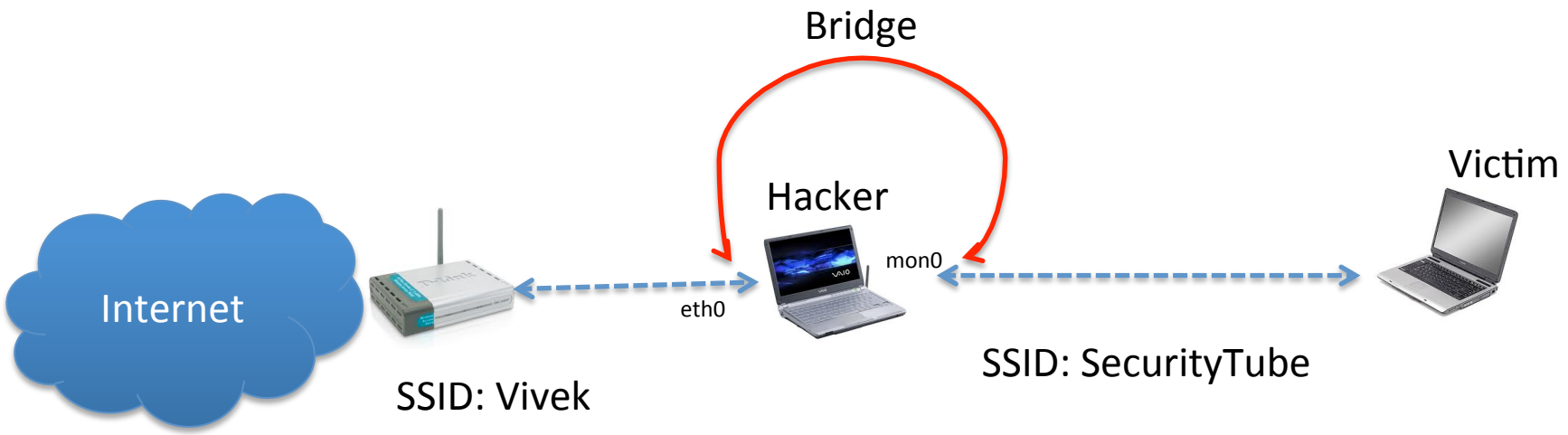
Our Setup



Understanding the Hack



SSL MITM



WEP Basics

- The first encryption scheme made available for Wi-Fi
- Flawed from the get go
- Uses RC4 encryption algorithm
  - Symmetric Key Encryption
- Is available on all access points
- Typically used by home users or manufacturing companies

WEP Internals

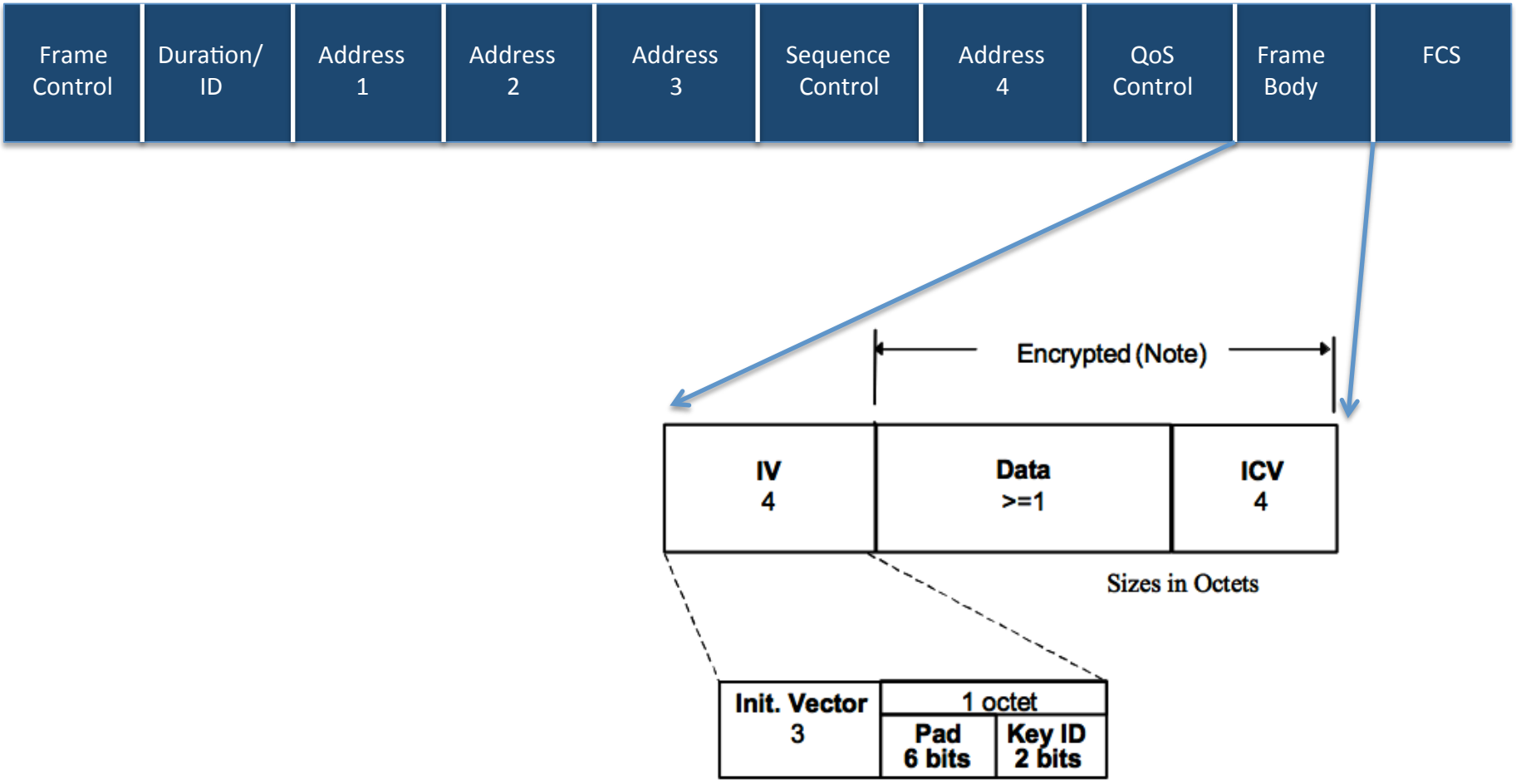
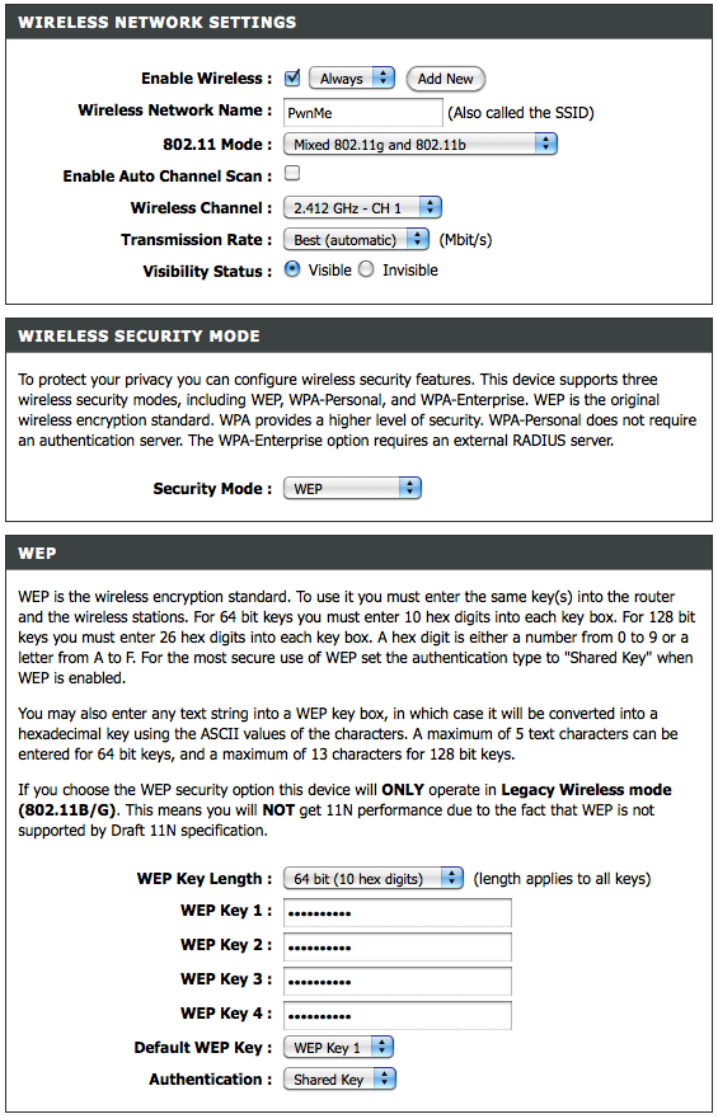


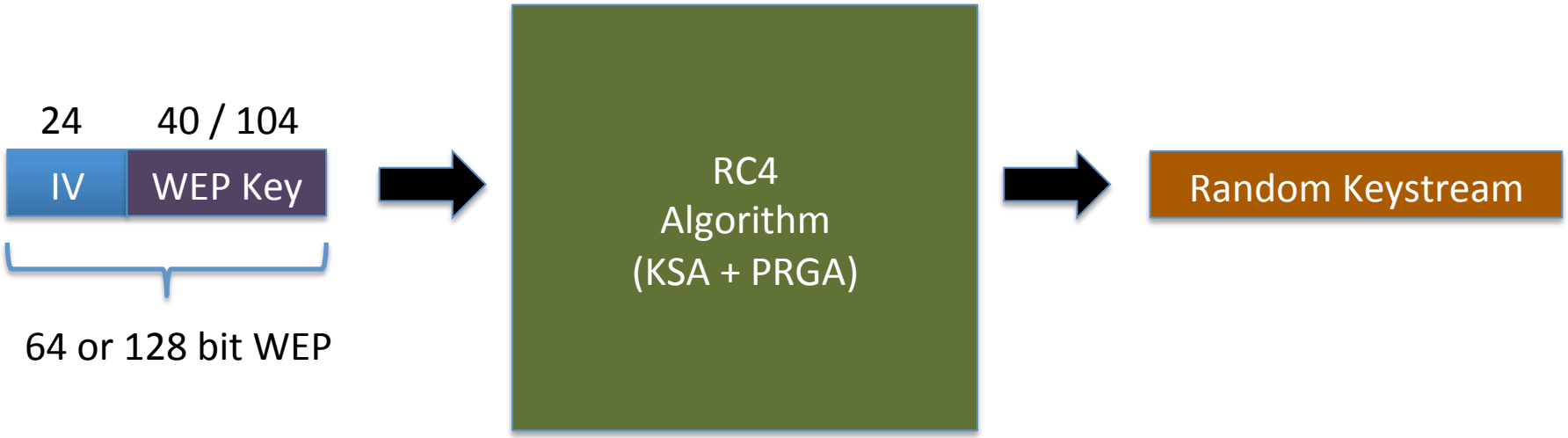
Figure 8-1—Construction of expanded WEP MPDU

Source: IEEE Standard

Multiple Keys

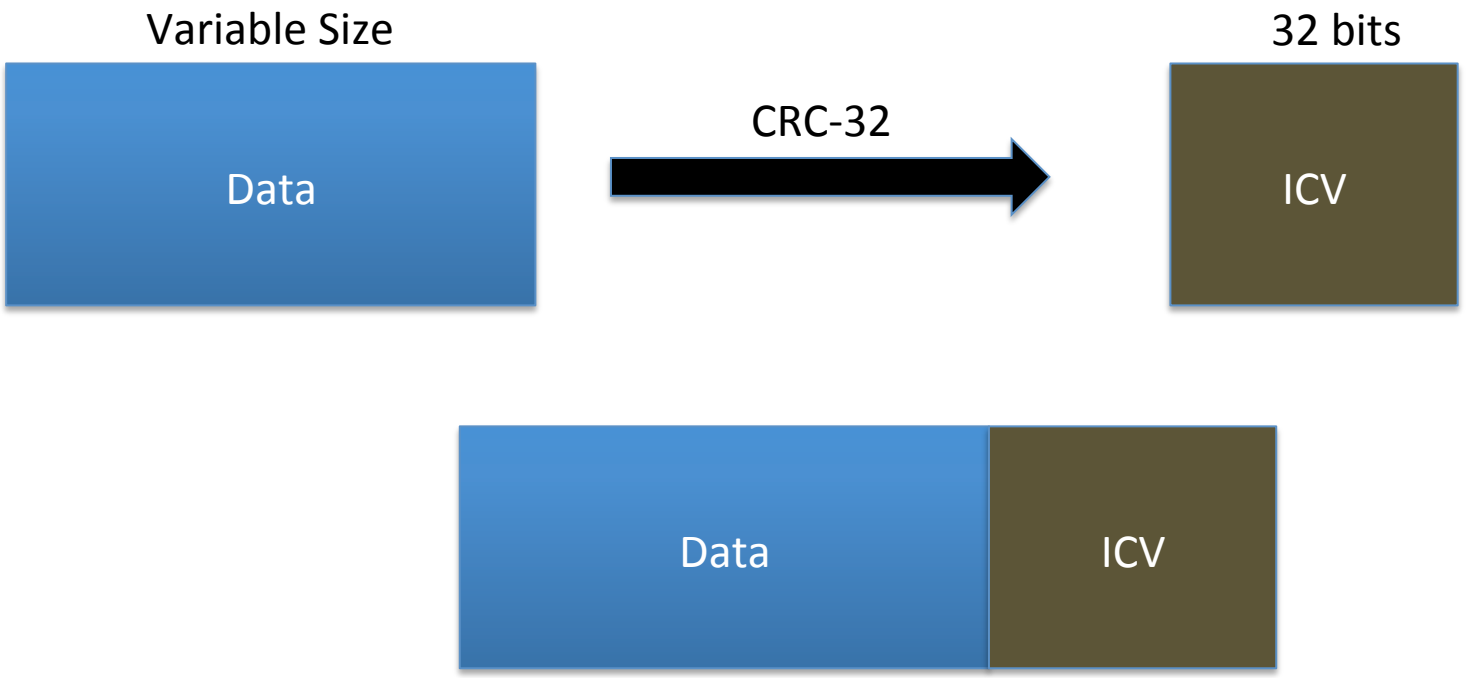


WEP Step 1: Generating the Keystream

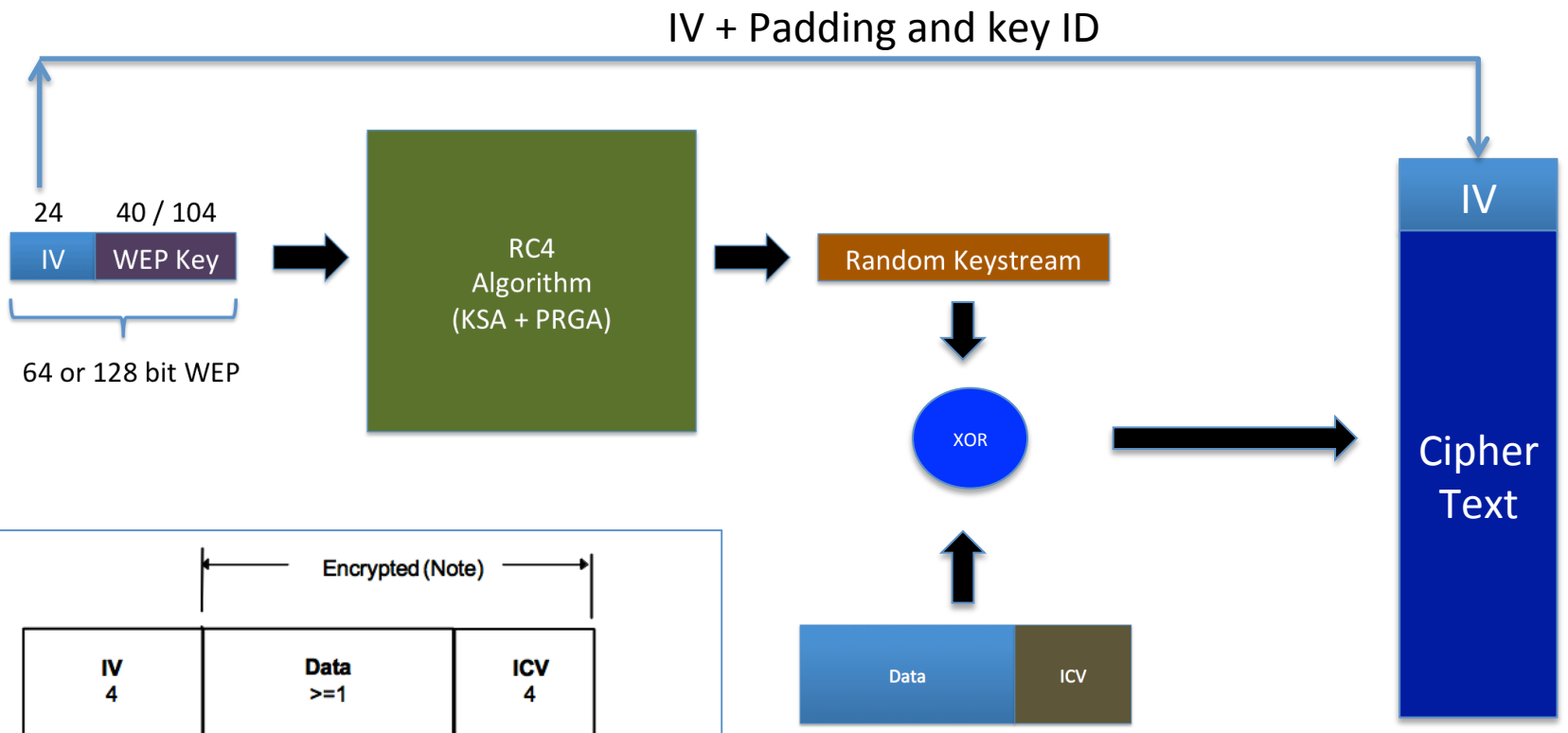


- RC4 Basics and Programming a simple RC4 Encrypt / Decrypt Software
- http://www.securitytube.net/video/38
- http://www.securitytube.net/video/79
- http://www.securitytube.net/video/40
- Basics C Programming Required

WEP Step 2: Generate Integrity Check Value



Step 3: Cipher Text Generation



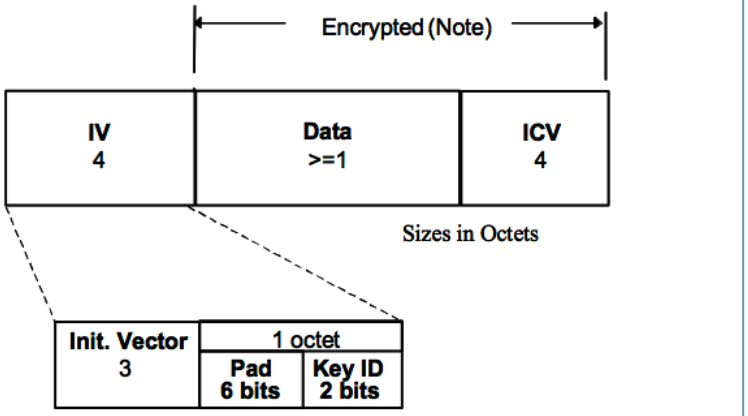


Figure 8-1—Construction of expanded WEP MPDU

IEEE Diagram for Encryption

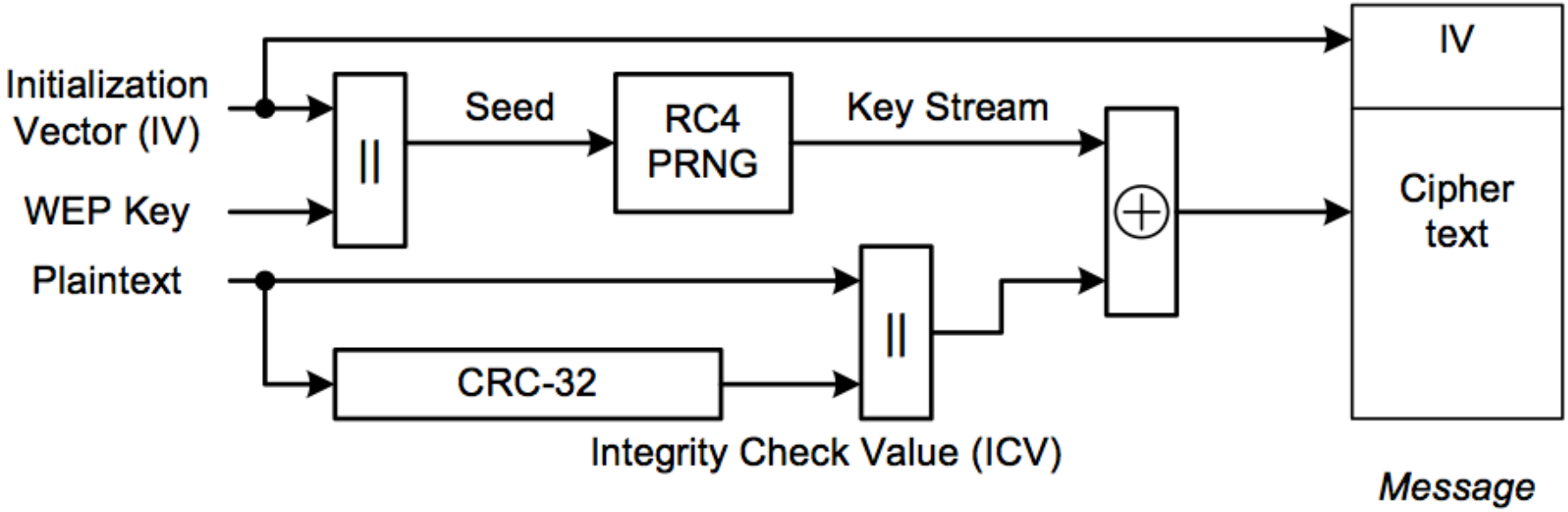


Figure 8-2—WEP encapsulation block diagram

WEP Internals

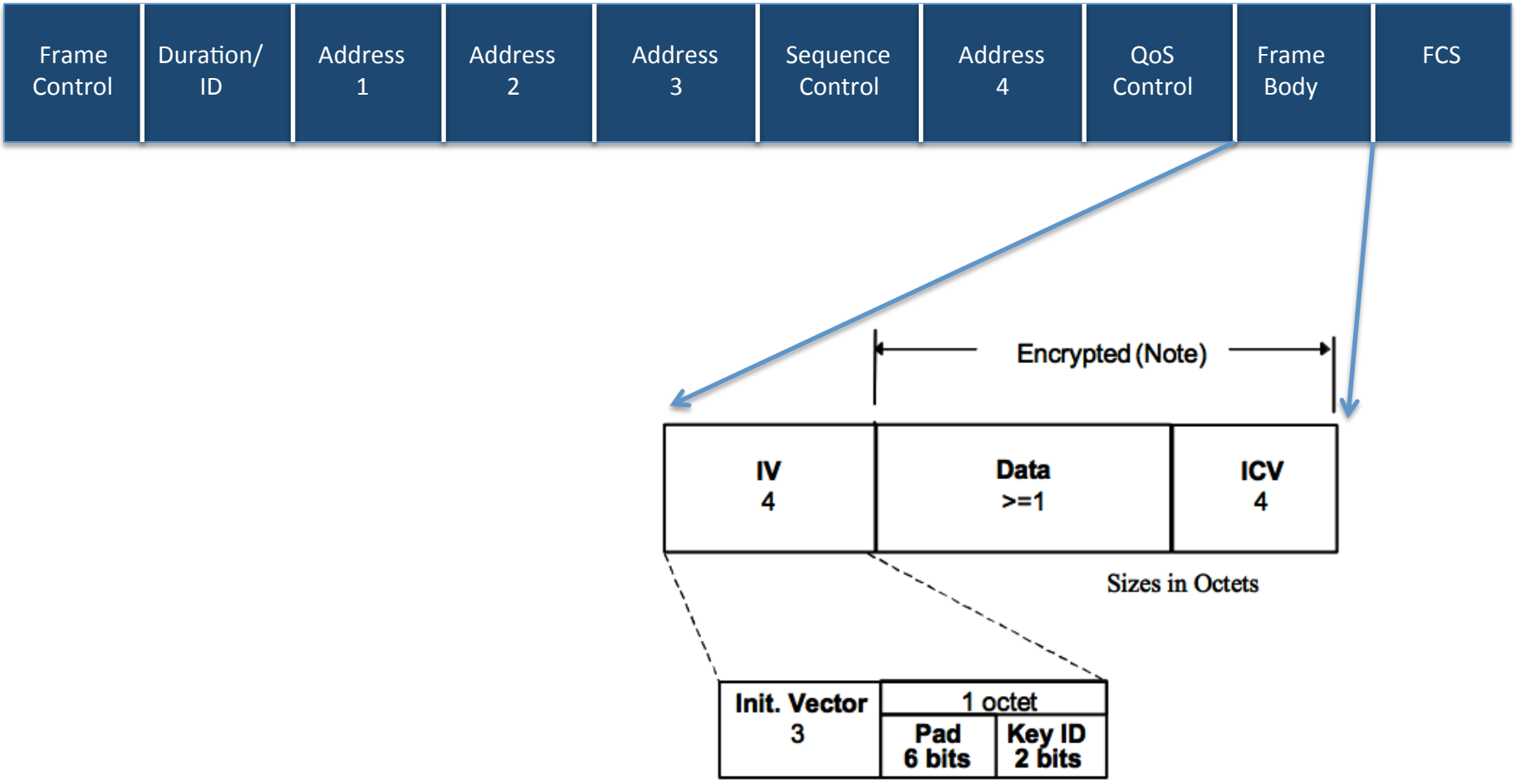
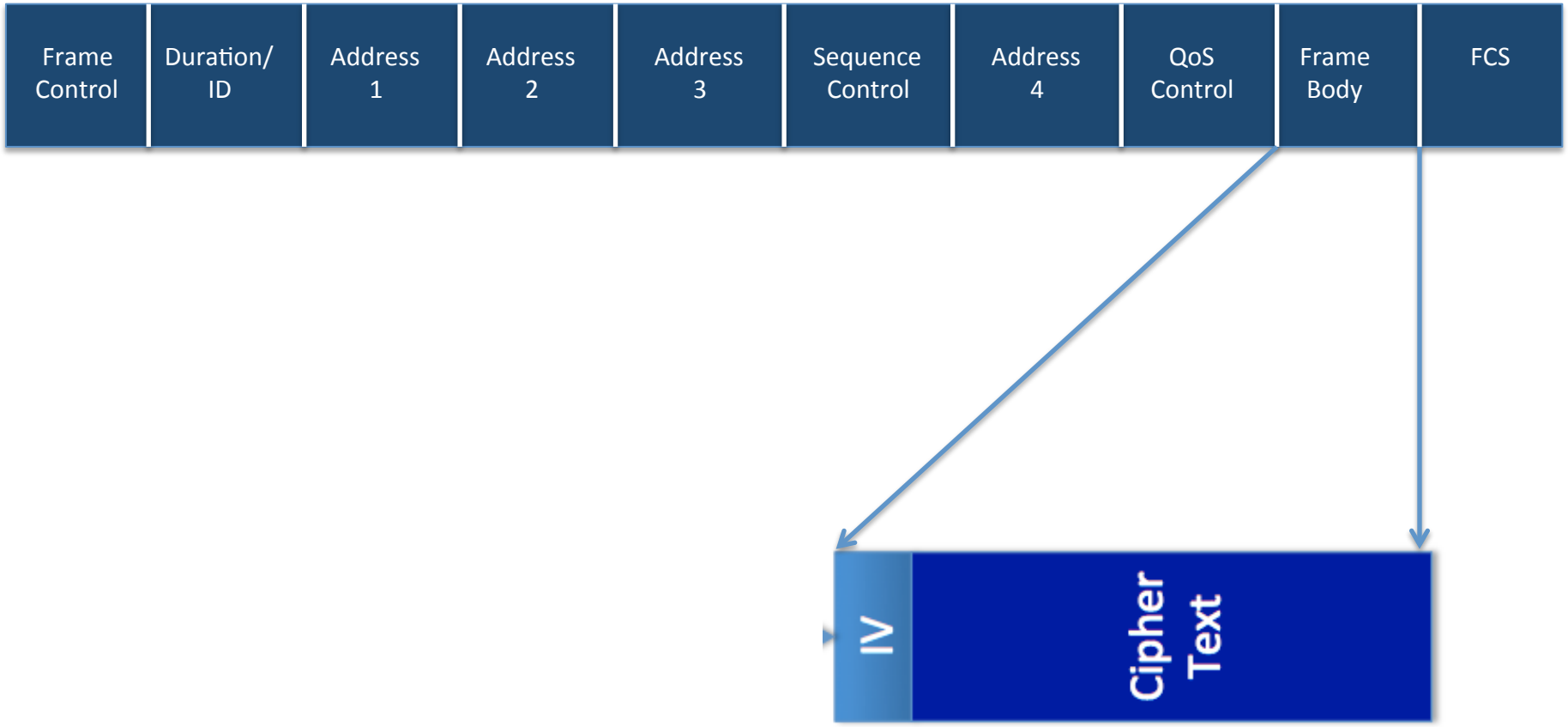


Figure 8-1—Construction of expanded WEP MPDU

Source: IEEE Standard

WEP Internals



Source: IEEE Standard

WEP Decryption

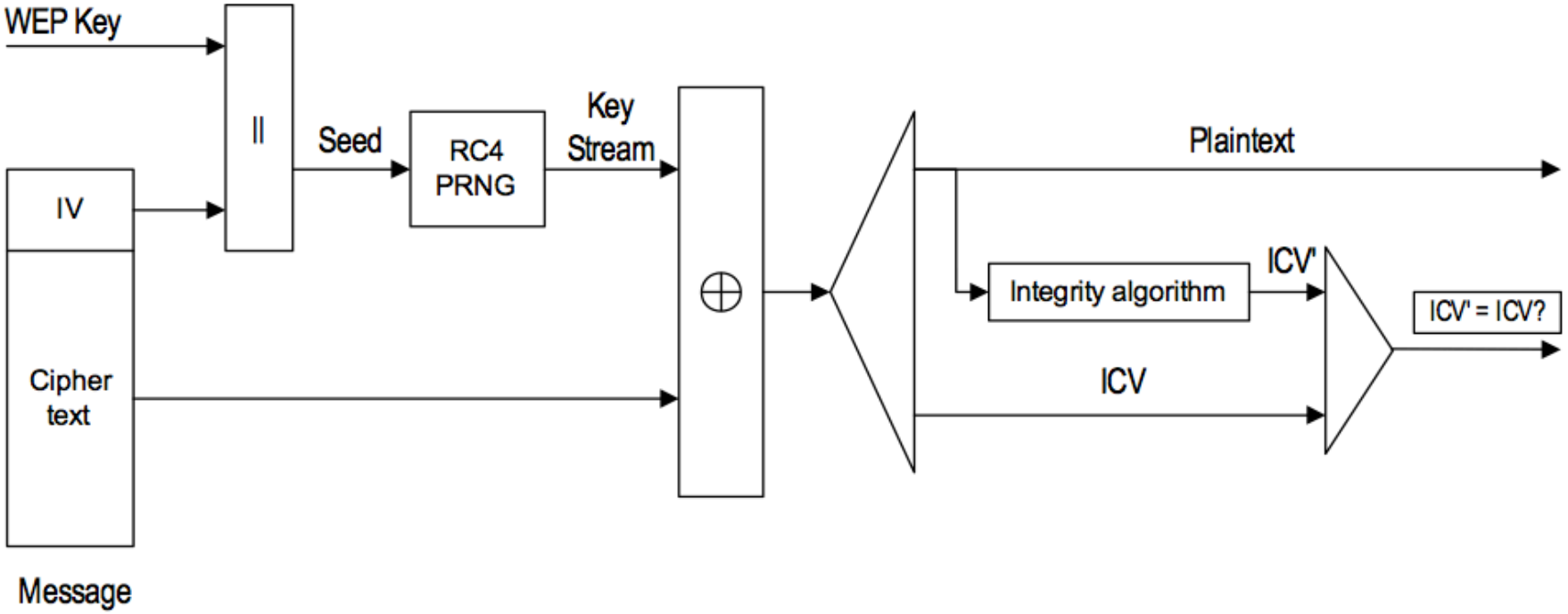


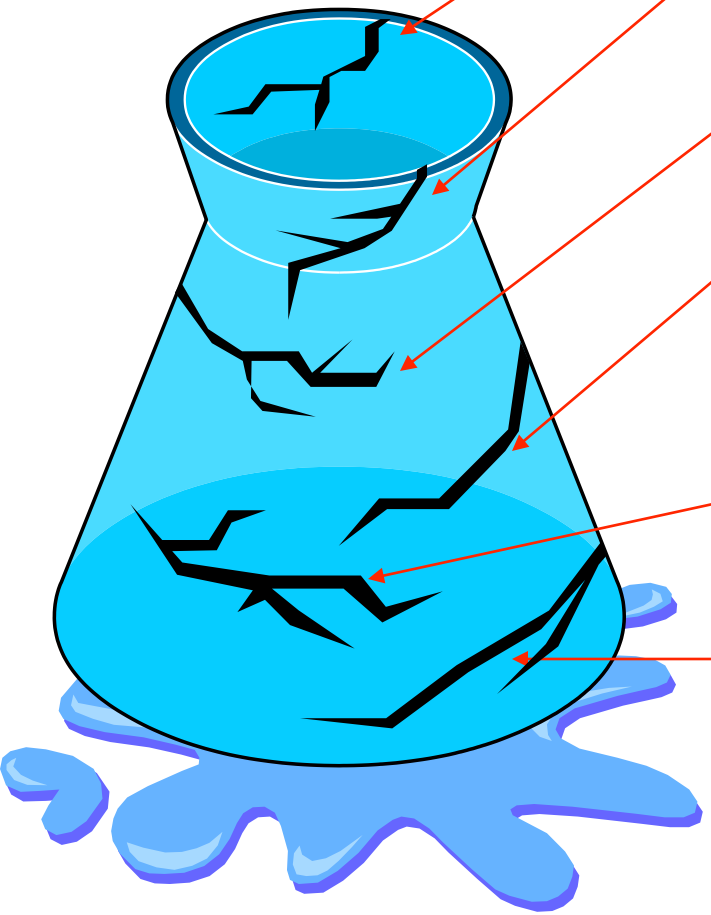
Figure 8-3—WEP decapsulation block diagram

Using Wireshark to Decrypt WEP

- Once we have the WEP key
  - Legitimate way
  - Or crack it ☺
- Airdecap-NG can also do the job

Broken Beyond Repair

IEEE WG admitted that WEP cannot hold any water. Recommended users to upgrade to WPA, WPA2



2001 - The insecurity of 802.11, Mobicom, July 2001
N. Borisov, I. Goldberg and D. Wagner.

2001 - Weaknesses in the key scheduling algorithm of RC4.
S. Fluhrer, I. Mantin, A. Shamir. Aug 2001.

2002 - Using the Fluhrer, Mantin, and Shamir Attack to Break WEP
A. Stubblefield, J. Ioannidis, A. Rubin.

2004 – KoreK, improves on the above technique and reduces the complexity of WEP cracking. We now require only around 500,000 packets to break the WEP key.

2005 – Adreas Klein introduces more correlations between the RC4 key stream and the key.

2007 – PTW extend Andreas technique to further simplify WEP Cracking. Now with just around 60,000 – 90,000 packets it is possible to break the WEP key.

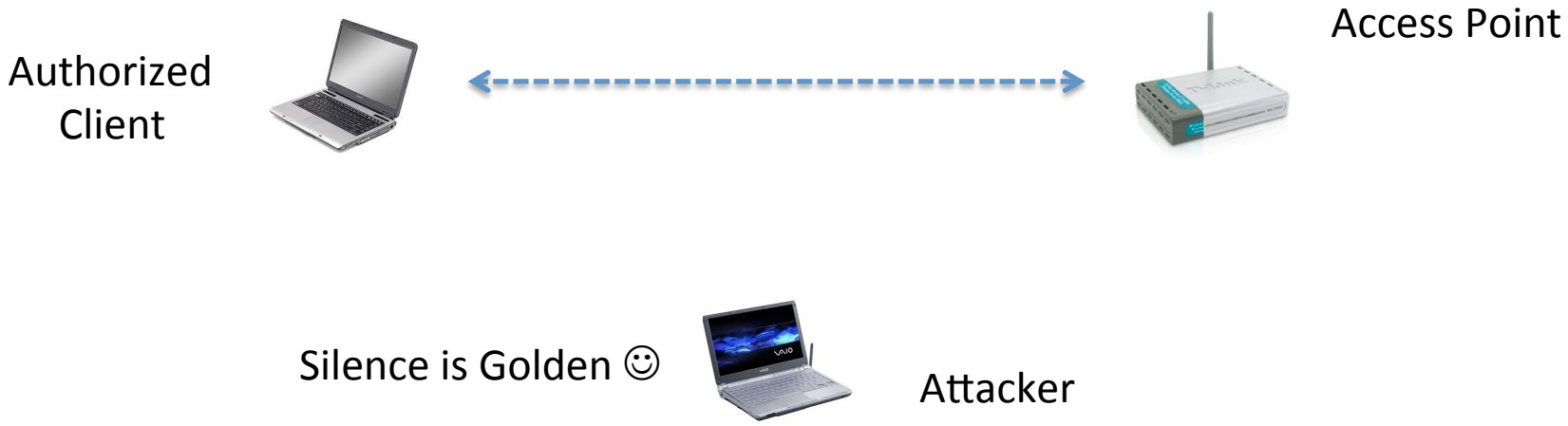
WEP Cracking

- Different Attacks using different logic
- Oldest one is finding “weak IVs” which reveal information about the WEP key
- Once you can collect a large number of weak IVs, you can crack the WEP key
- Weak IVs are not uniformly distributed in the IV space
- A Weak IV is key dependent
- This is the reason why it takes some time

Cracking WEP – the script kiddie way ☺

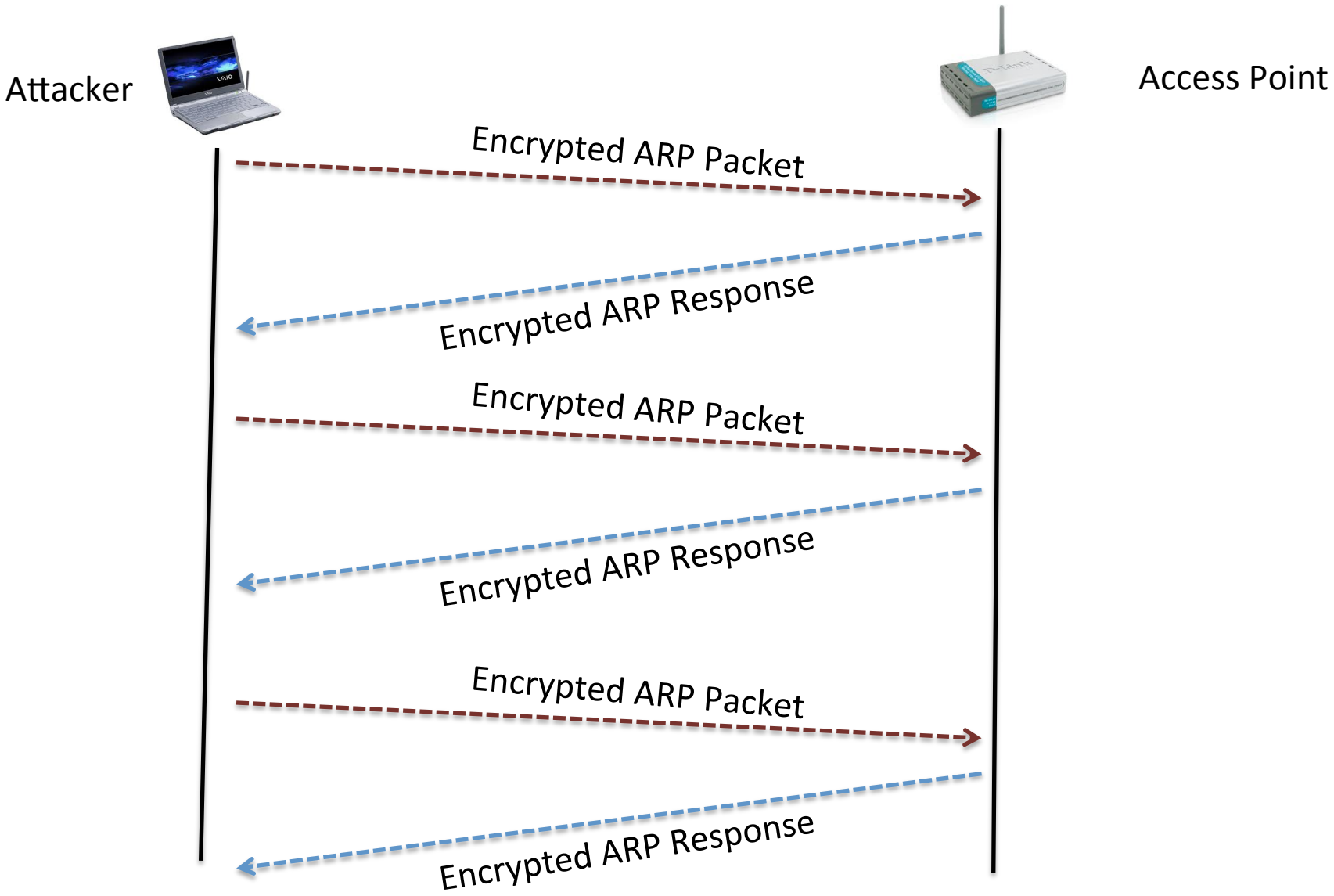
- Techniques
  - Passive Way (Wait … wait … wait)
    - Advantage – Undetectable
    - Use Directional Antenna
    - Decrypt traffic once cracked
  - Active Way (Patience is not your virtue)
    - Replay attacks
      - Stimulate the network to send encrypted data packets
    - ARP Replay
      - ARP Request, sends ARP Response

ARP Replay Step 1: Capture ARP Packets



- How does the Attacker Identify the ARP Packets? Aren't they all encrypted?
- ARP packets are of a fixed unique size, easy to identify even if encrypted
- Capture ARP Request packets using encrypted packet size and Destination MAC address
- Replay them blindly, and see if the network responds back!
- If yes, then we found ourselves Winner ☺ ☺

ARP Replay Step 2: Replay Packets to AP



ARP Replay Step 3: Collect Packets and use Aircrack-NG

```
                                Aircrack-ng 1.1 r1738

                        [00:00:00] Tested 7 keys (got 23709 IVs)

KB    depth   byte(vote)
 0    1/  2   AB(31744) 8D(30976) 16(30464) CE(30464) 6C(29952) 28(29696) 5D(28672) 63(28672) 47(28416)
 1    0/  3   26(31744) 57(30720) 78(30464) 41(29184) D2(29184) DC(28928) E7(28928) F4(28928) A2(28672)
 2    0/  1   EF(41472) 6B(30976) 86(29952) 30(29696) 3C(29184) 62(29184) 8C(29184) 98(29184) B4(28672)
 3    0/  1   AB(34304) 5B(29184) 8E(29184) 56(28672) 0D(28416) 24(28416) 3F(28416) 9F(28416) B8(28416)
 4    0/  1   CD(34304) F0(31488) D7(29184) A1(28928) 88(28672) A0(28416) DA(28416) CA(28160) D8(28160)

                    KEY FOUND! [ AB:CD:EF:AB:CD ]
     Decrypted correctly: 100%
```

Its not just the Encryption

- Message Injection Attacks
  - No replay protection
  - Aireplay-NG ARP Replay
- Message Injection
  - ChopChop attack
  - Caffe latte attack
  - Fragmentation attack
  - Hirte Attack

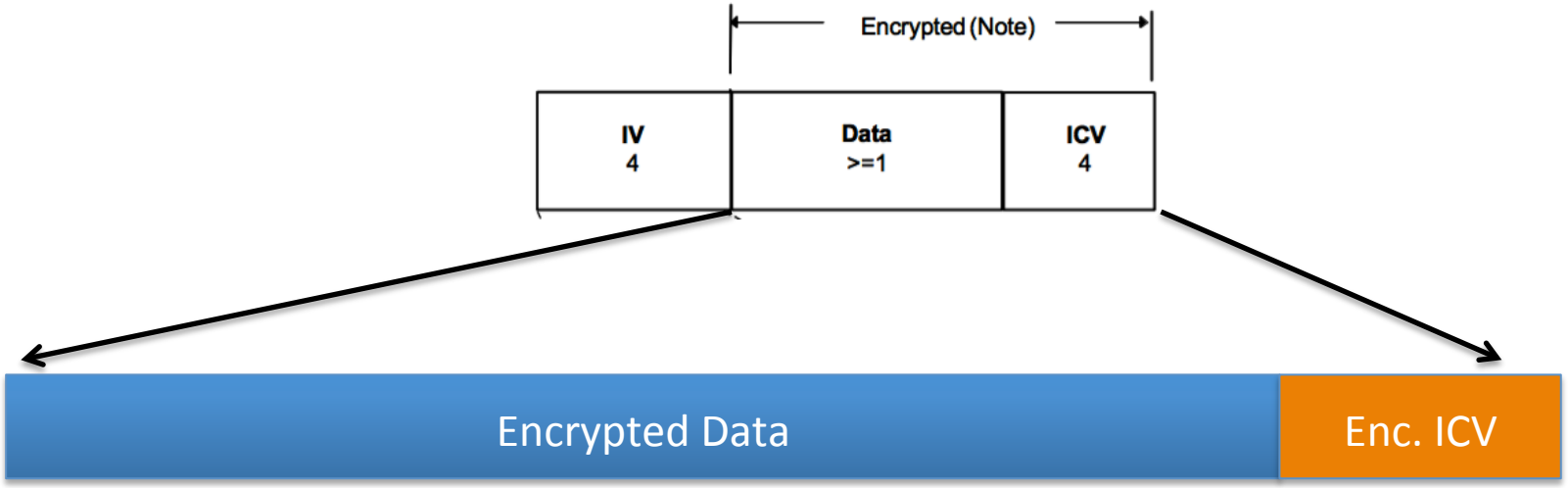
Message Modification

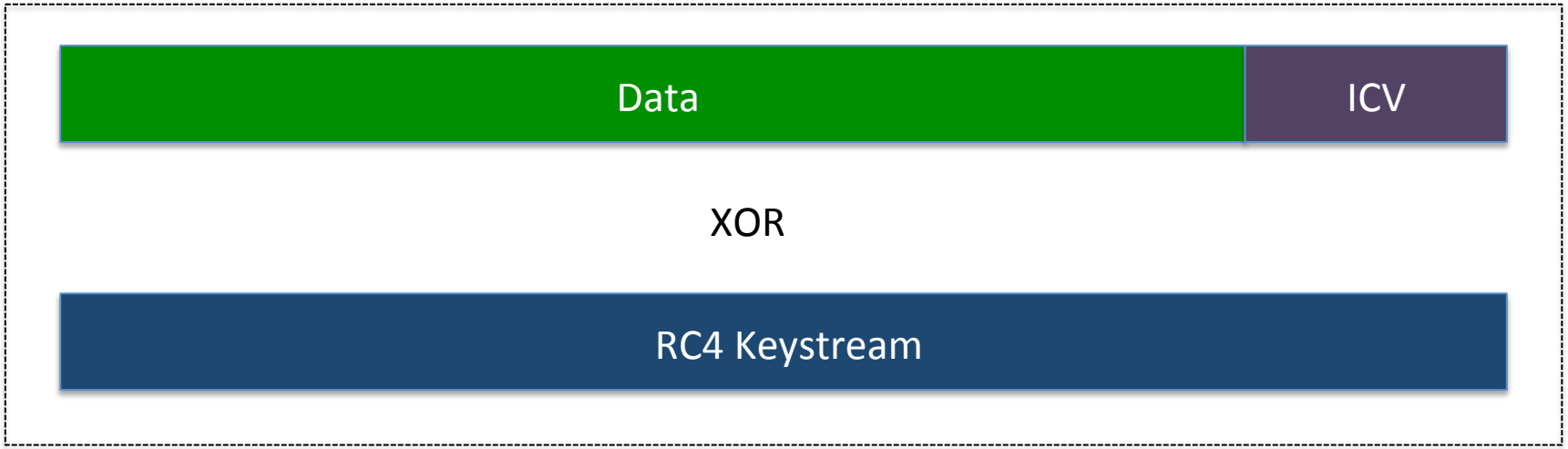
$$
\begin{aligned}
C' &= C \oplus \langle \Delta, c(\Delta) \rangle \\
&= \mathrm{RC4}(v,k) \oplus \langle M, c(M) \rangle \oplus \langle \Delta, c(\Delta) \rangle \\
&= \mathrm{RC4}(v,k) \oplus \langle M \oplus \Delta, c(M) \oplus c(\Delta) \rangle \\
&= \mathrm{RC4}(v,k) \oplus \langle M', c(M \oplus \Delta) \rangle \\
&= \mathrm{RC4}(v,k) \oplus \langle M', c(M') \rangle .
\end{aligned}
$$

- CRC-32 is a linear function of the message, hence checksum is distributive over XOR
- Thus we can tamper arbitrary byte locations in the packet and patch the checksum
- This will be a valid packet accepted by the access point

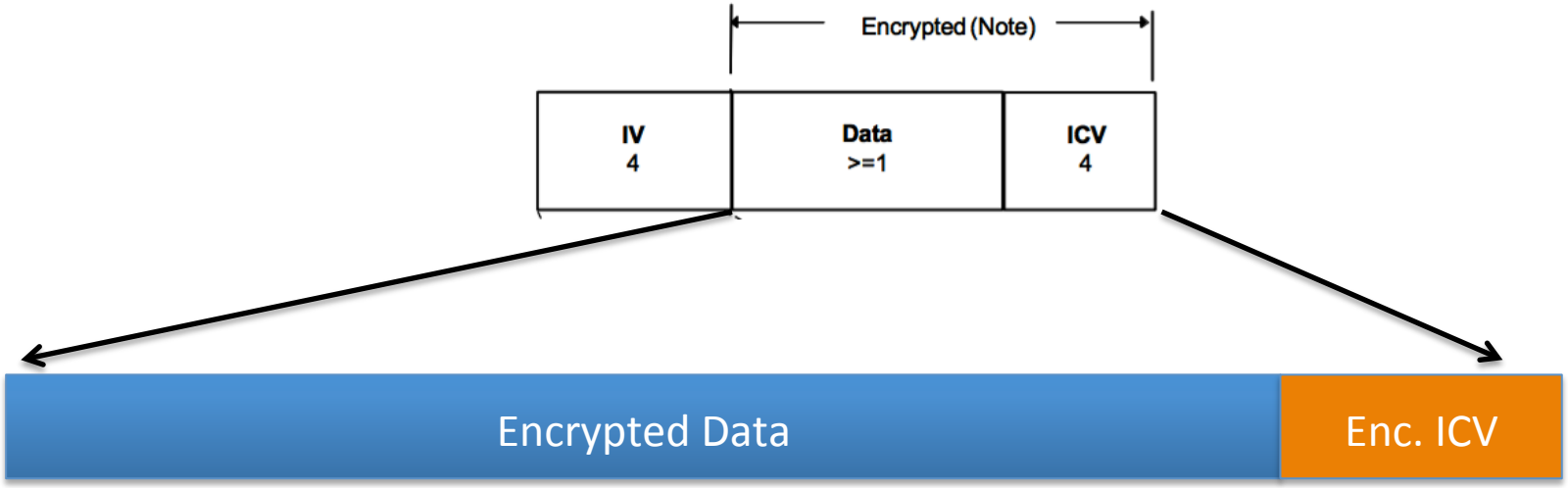
Original Research Paper: *Intercepting Mobile Communications*

What does this mean for us?





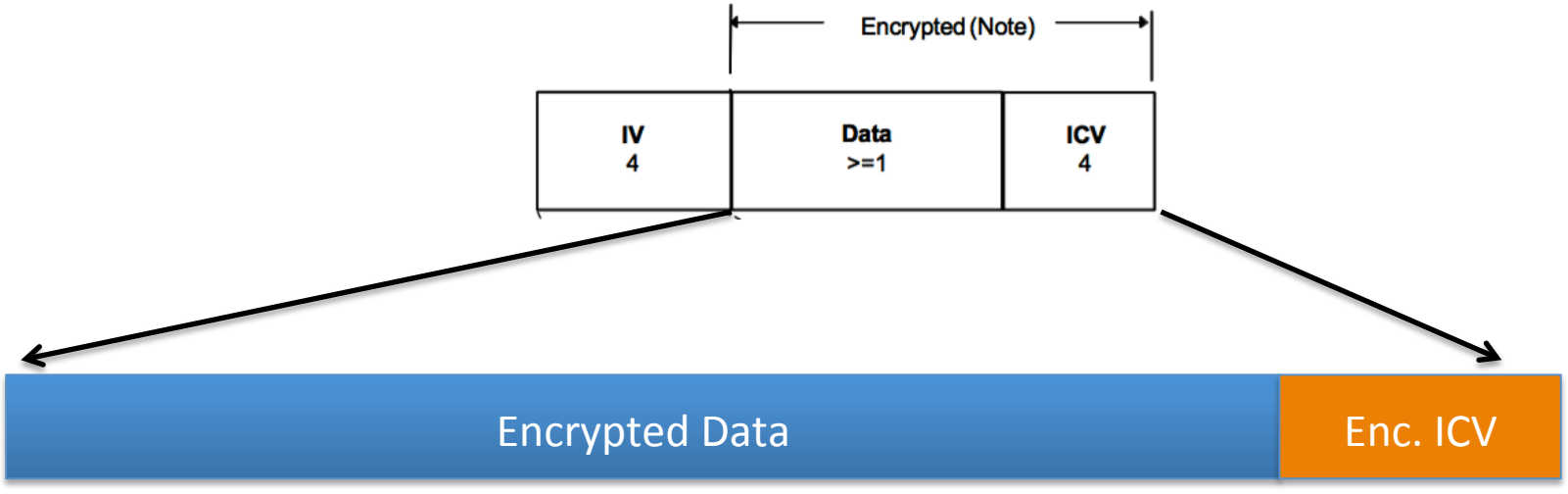
Create a Bit Mask without knowing Plain Text



Attacker Modifications

ICV Patch = CRC-32 of Attacker Modifications

Patching a Valid Packet



Encrypted Data | Enc. ICV

XOR

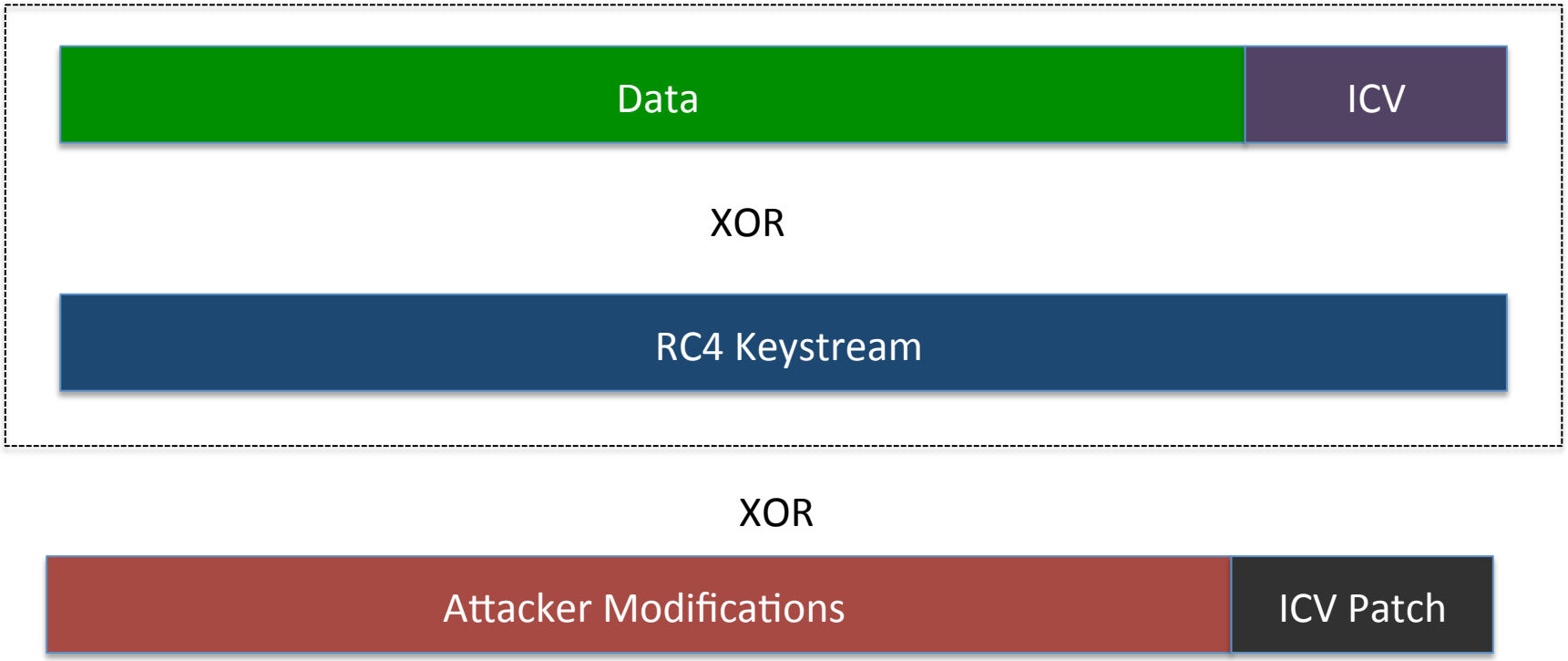
Attacker Modifications | ICV Patch

||

Modified Encrypted Data | Enc. ICV

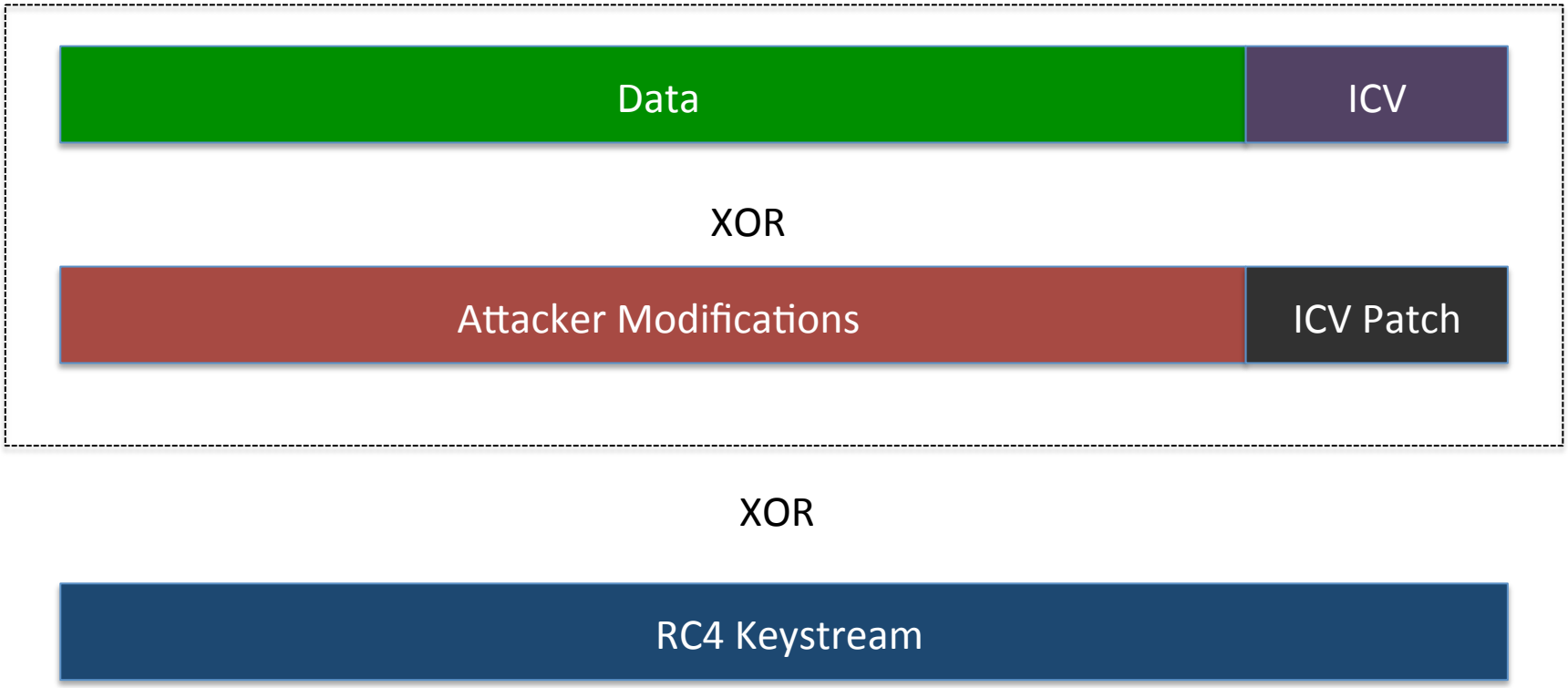
Valid Encrypted Packet!

Behind the Scenes

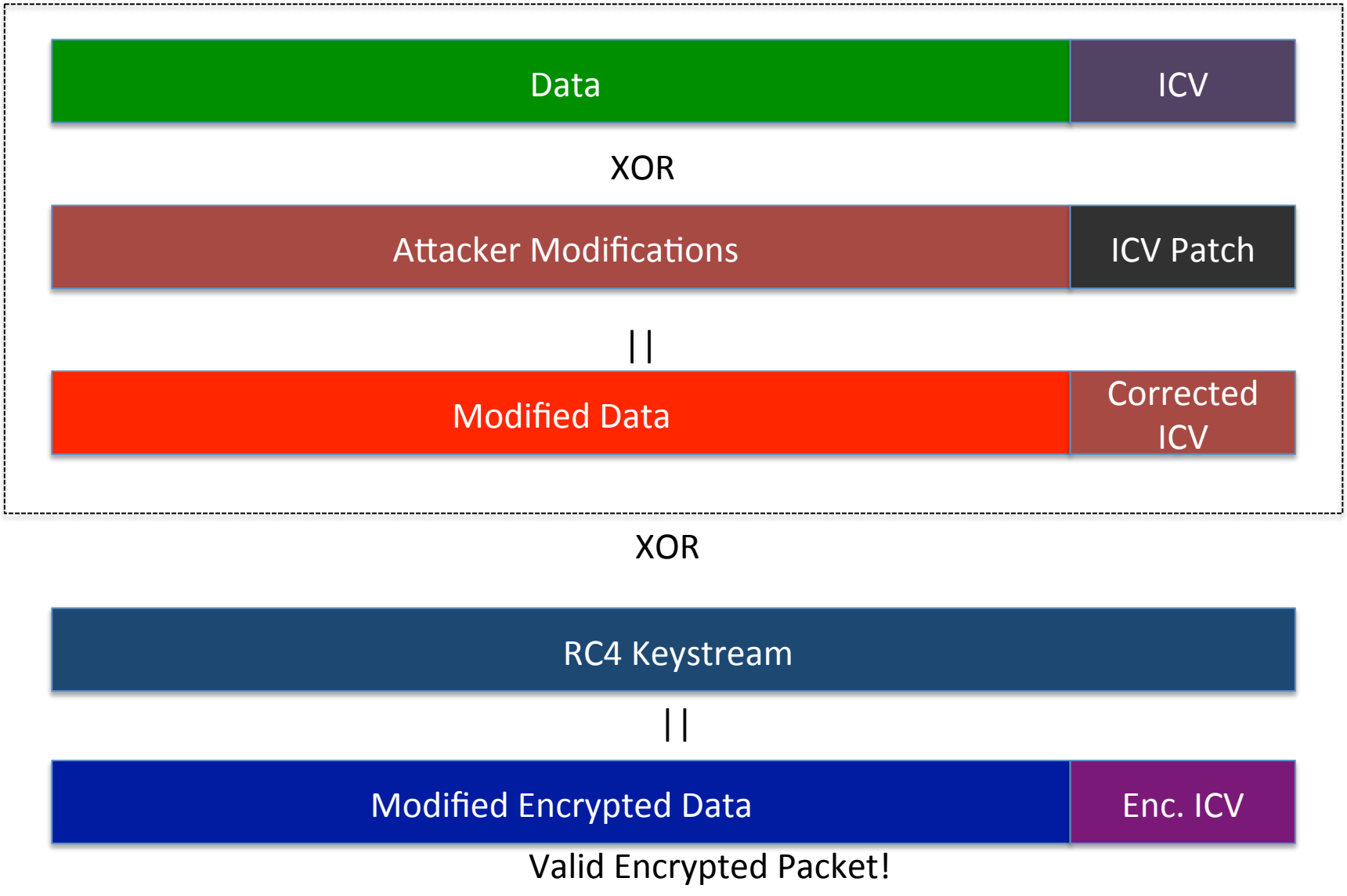


A (xor) B (xor) C = [A (xor) C] (xor) B

A (xor) B (xor) C = [A (xor) C] (xor) B



Modified packet XOR with Keystream



Repercussions

- We can modify arbitrary data in a WEP packet and patch the ICV
- This is a valid WEP packet which will be accepted by the Access Point / client

- Caffe Latte attack Modifies a Gratuitous ARP packet to change it to a ARP Request packet for the same host!
- Host Replies and we collect these packets to crack the WEP key

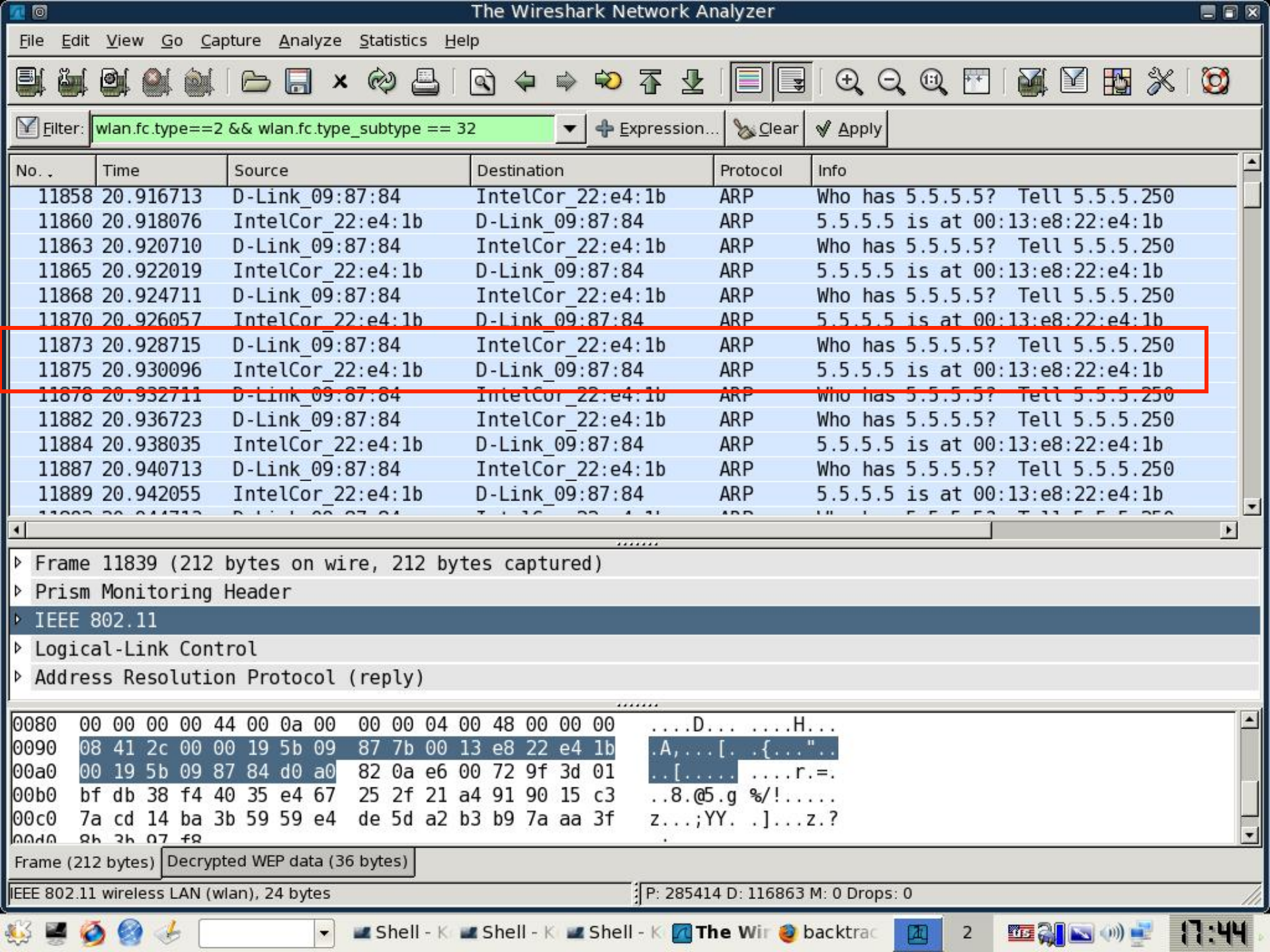
A Cup of Caffe Latte served with the WEP key! ☺

Images copyright Airtight Networks

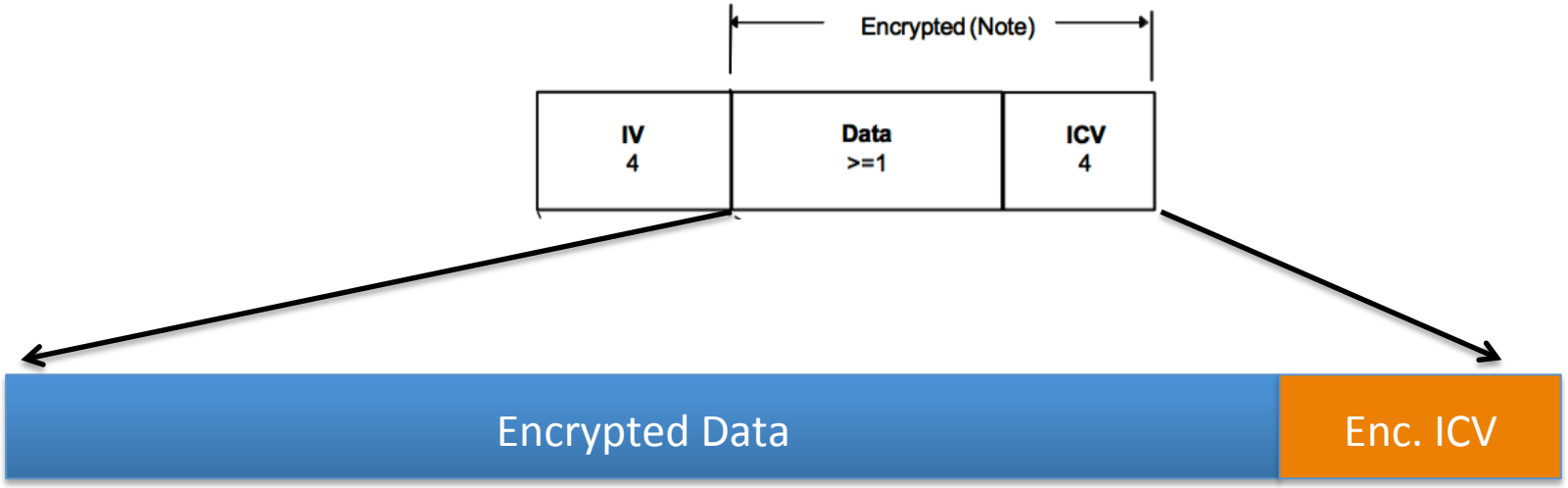
Caffe Latte Details

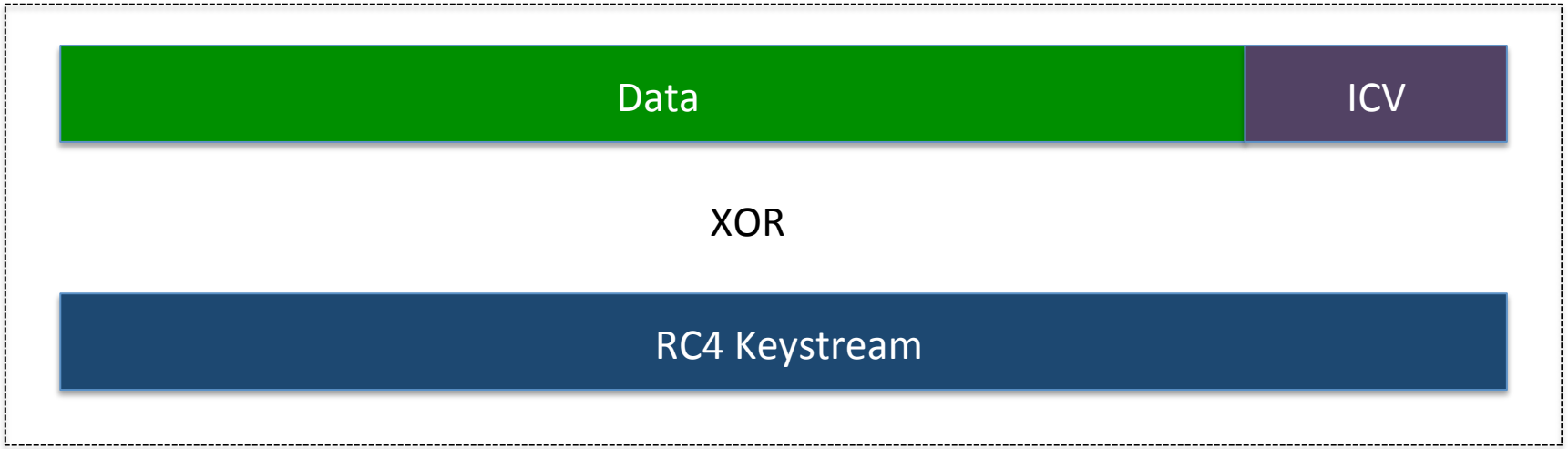
- Once the client connects to the fake AP it will send out DHCP requests
- DHCP will time out eventually
- Auto-configuration IP address will kick in
- Client will send a Gratuitous ARP packet

Let us Verify!


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Help
Filter: wlan.fc.type==2 && wlan.fc.type_subtype == 32
Expression... Clear Apply
No. Time Source Destination Protocol Info
11858 20.916713 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11860 20.918076 IntelCor_22:e4:1b D-Link_09:87:84 ARP 5.5.5.5 is at 00:13:e8:22:e4:1b
11863 20.920710 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11865 20.922019 IntelCor_22:e4:1b D-Link_09:87:84 ARP 5.5.5.5 is at 00:13:e8:22:e4:1b
11868 20.924711 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11870 20.926057 IntelCor_22:e4:1b D-Link_09:87:84 ARP 5.5.5.5 is at 00:13:e8:22:e4:1b
11873 20.928715 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11875 20.930096 IntelCor_22:e4:1b D-Link_09:87:84 ARP 5.5.5.5 is at 00:13:e8:22:e4:1b
11878 20.932711 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11882 20.936723 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11884 20.938035 IntelCor_22:e4:1b D-Link_09:87:84 ARP 5.5.5.5 is at 00:13:e8:22:e4:1b
11887 20.940713 D-Link_09:87:84 IntelCor_22:e4:1b ARP Who has 5.5.5.5? Tell 5.5.5.250
11889 20.942055 IntelCor_22:e4:1b D-Link_09:87:84 ARP 5.5.5.5 is at 00:13:e8:22:e4:1b
Frame 11839 (212 bytes on wire, 212 bytes captured)
Prism Monitoring Header
IEEE 802.11
Logical-Link Control
Address Resolution Protocol (reply)
0080 00 00 00 00 44 00 0a 00 00 00 04 00 48 00 00 00 ....D... ....H...
0090 08 41 2c 00 00 19 5b 09 87 7b 00 13 e8 22 e4 1b .A,...[. .{..."..
00a0 00 19 5b 09 87 84 d0 a0 82 0a e6 00 72 9f 3d 01 ..[..... ....r.=.
00b0 bf db 38 f4 40 35 e4 67 25 2f 21 a4 91 90 15 c3 ..8.@5.g %/!.....
00c0 7a cd 14 ba 3b 59 59 e4 de 5d a2 b3 b9 7a aa 3f z...;YY. .]...z.?
Frame (212 bytes)
Decrypted WEP data (36 bytes)
IEEE 802.11 wireless LAN (wlan), 24 bytes
P: 285414 D: 116863 M: 0 Drops: 0
Shell - K
Shell - K
Shell - K
The Wir
backtrac
2
17:44

Back to the Drawing Board





Korek's ChopChop

Guess	New ICV	Accepted
00	ICV-1	No
01	ICV-2	No
…	…	…
FA	ICV-n	Yes!

FA

ChopChop

Guess	New ICV	Accepted
00	ICV-1	No
01	ICV-2	No
…	…	…
CD	ICV-n	Yes!

CD FA

End Result

- Decrypt entire WEP packet byte by byte
- Can be orchestrated in 2 modes:
  - Authenticated to AP
    - Packet is replayed by the AP over the air
  - Unauthenticated to AP
    - Some APs send a de-authentication packet if the WEP packet is valid but MAC is not associated
    - May not work always

Understanding Fragmentation

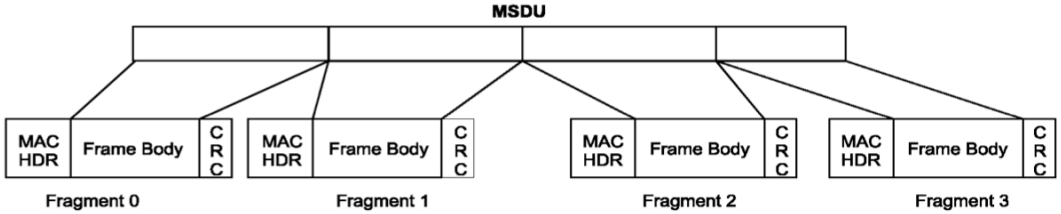
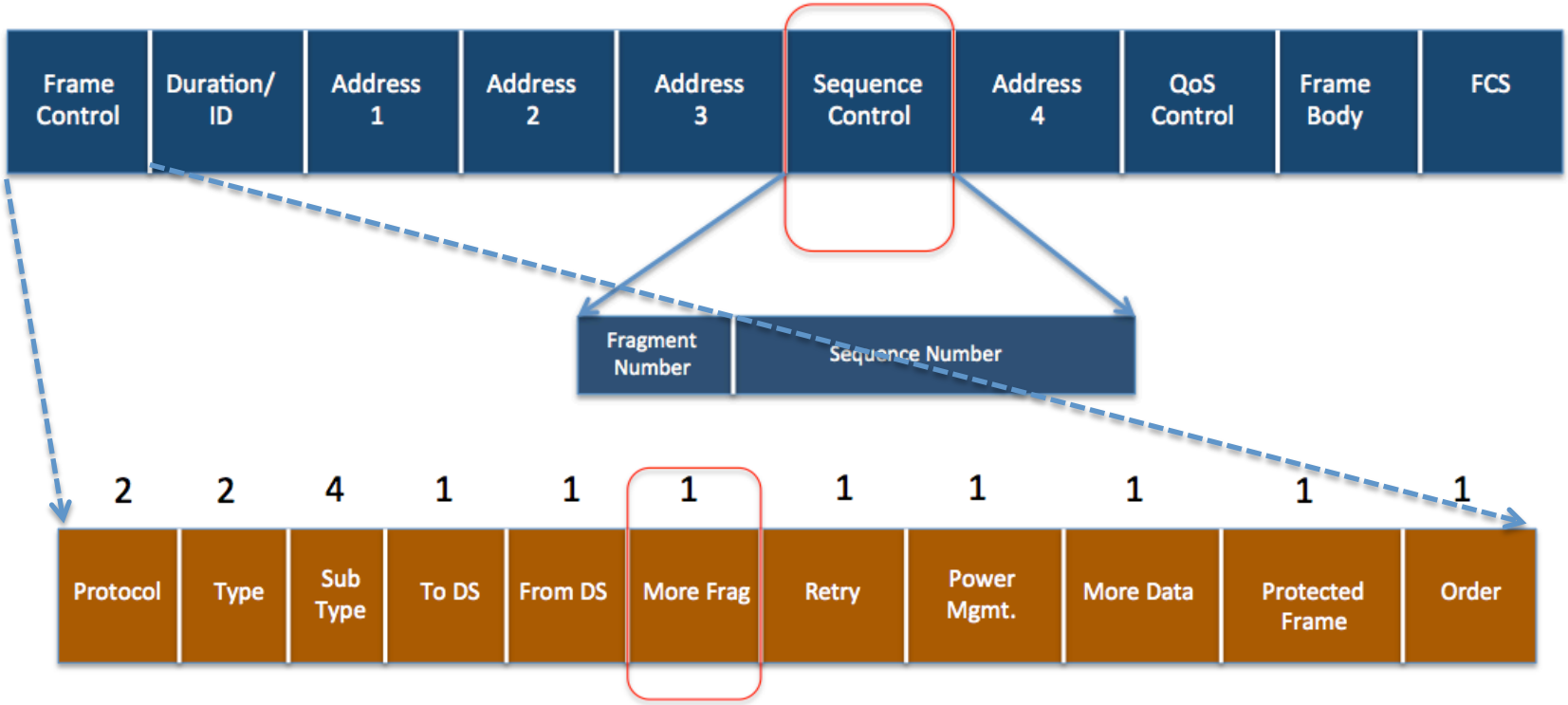
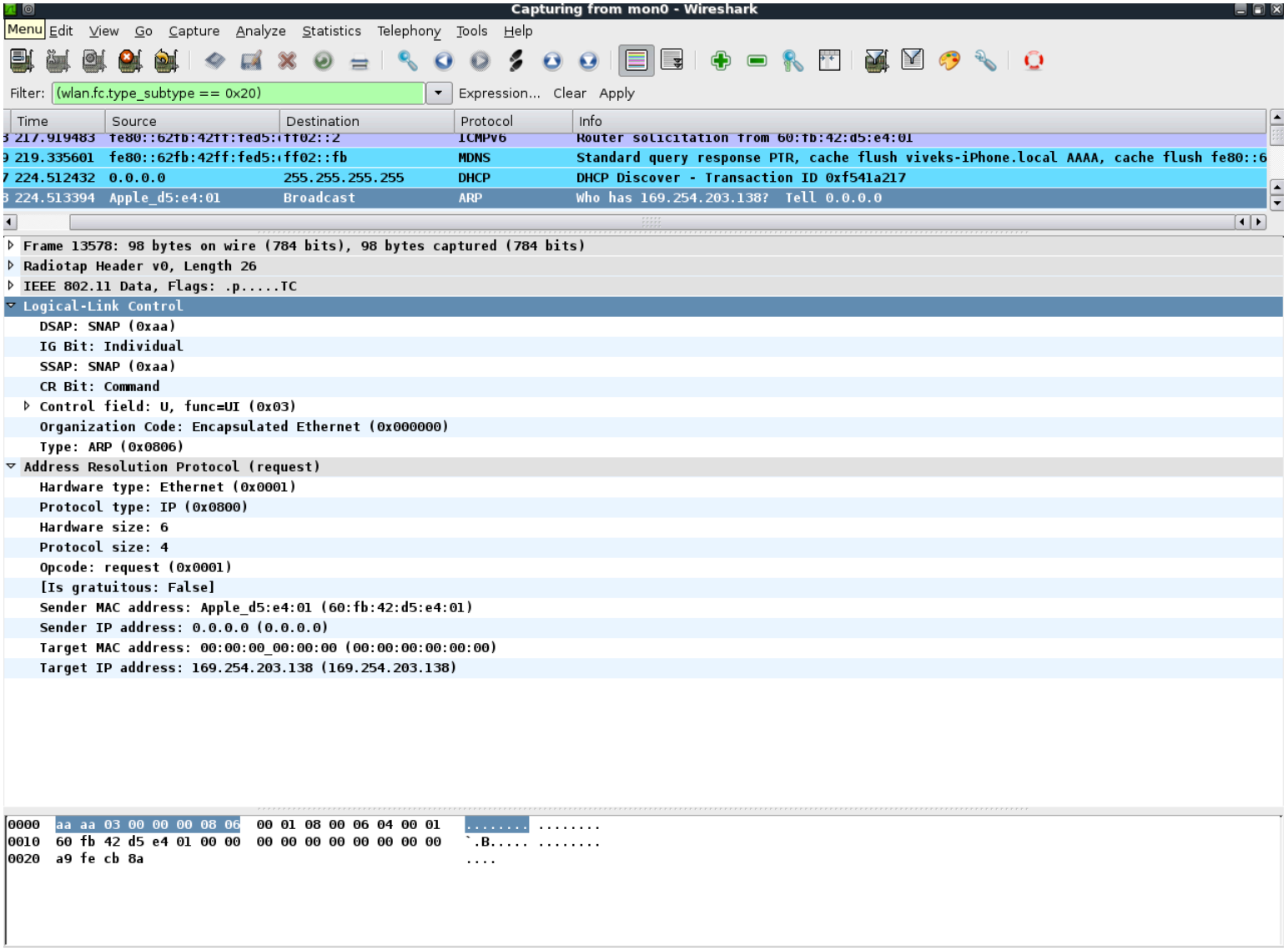


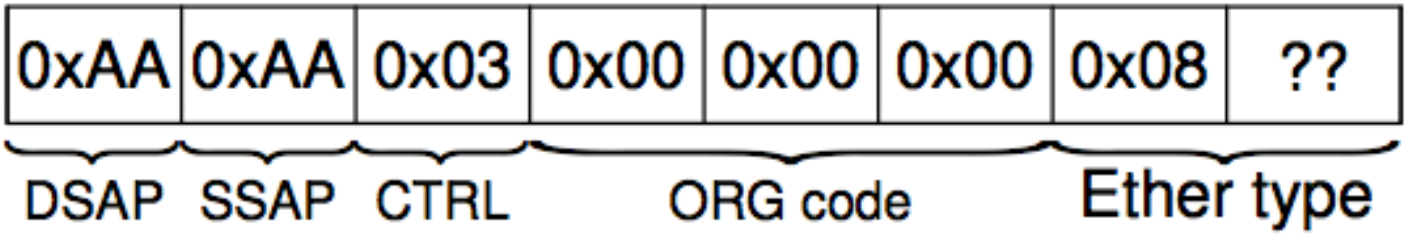
Figure 9-2—Fragmentation



LLC Header + Rest

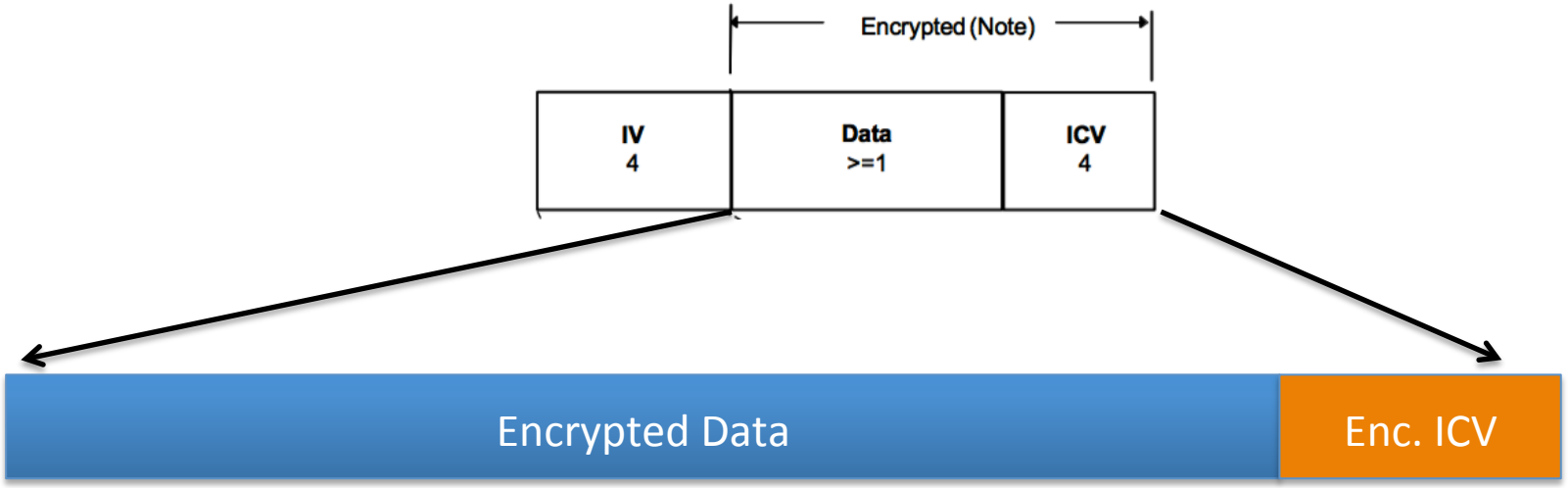


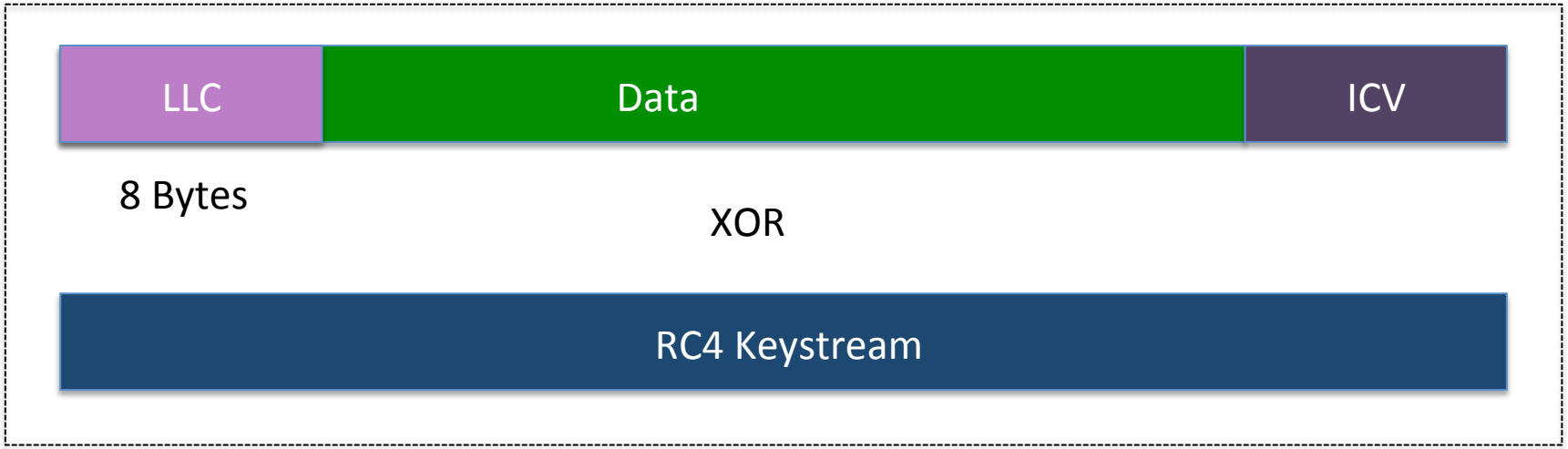
LLC Header is Known



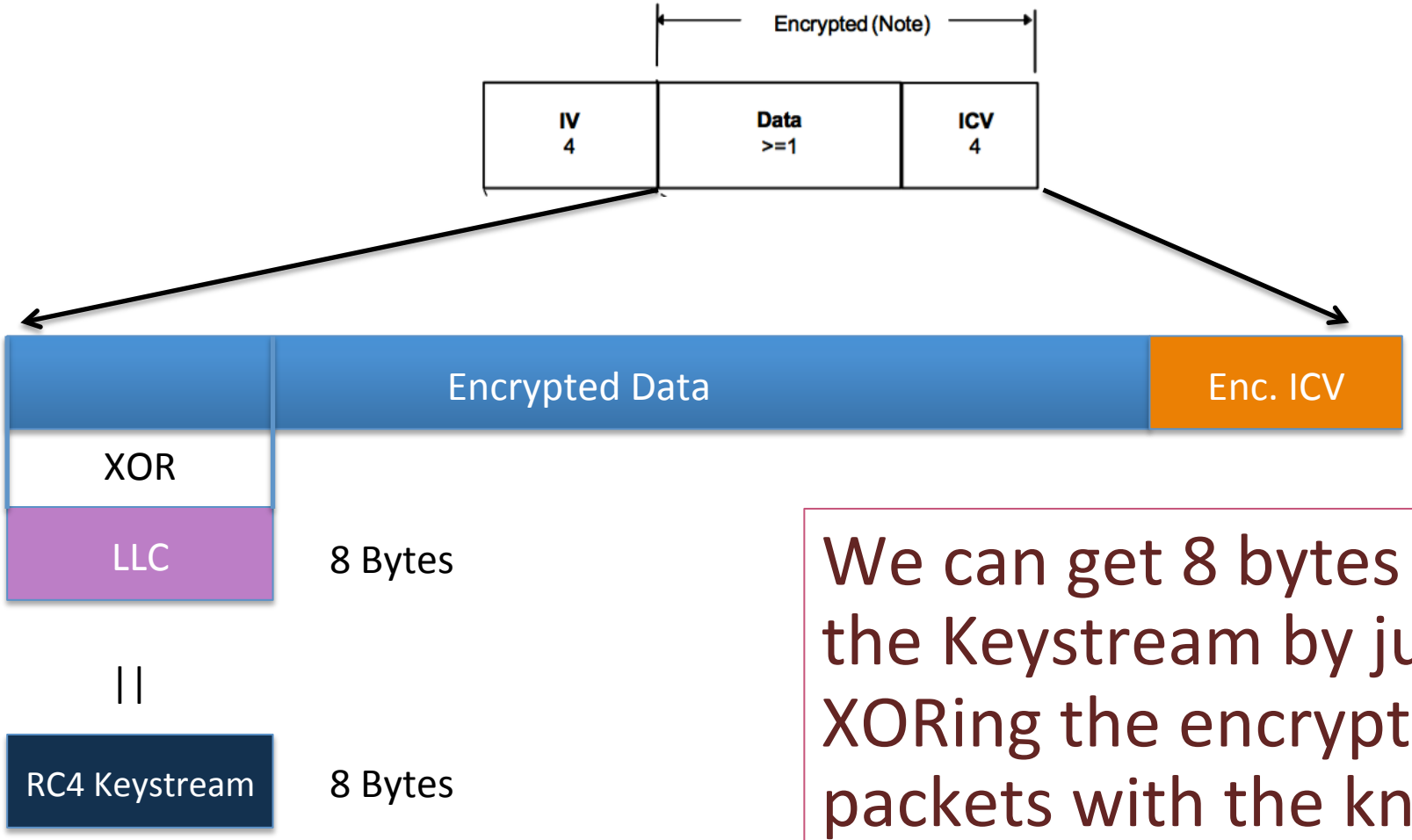
- 8 Bytes of LLC header is known
- Ether Type can be ARP / IP typically
- Can be guessed from the packet size

Packet Breakup





Known Plain Text Attack



We can get 8 bytes of the Keystream by just XORing the encrypted packets with the known plain text of the LLC

What do we have now?

RC4 Keystream

8 Bytes of Keystream + Corresponding IV

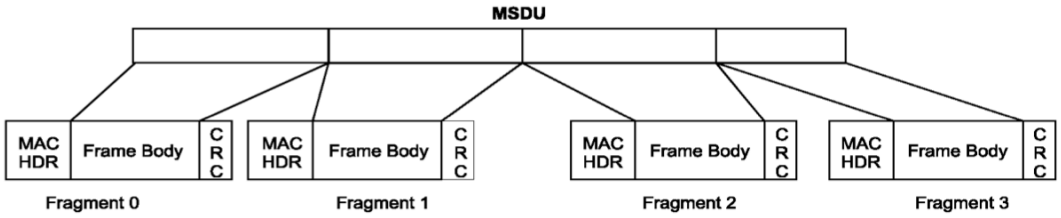
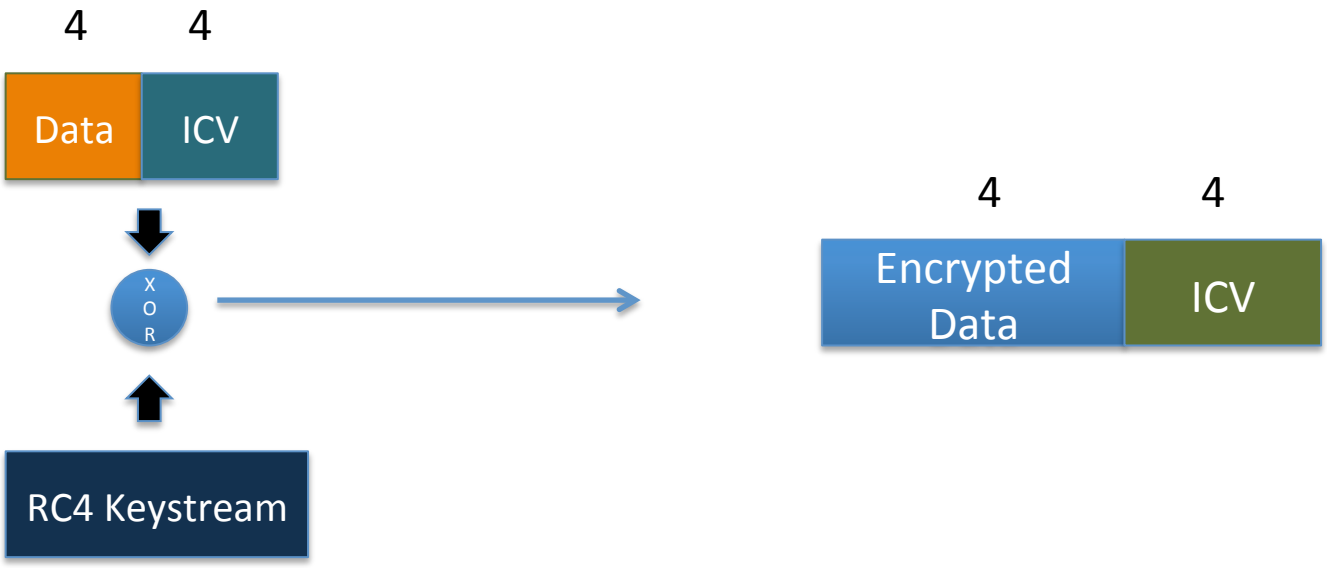
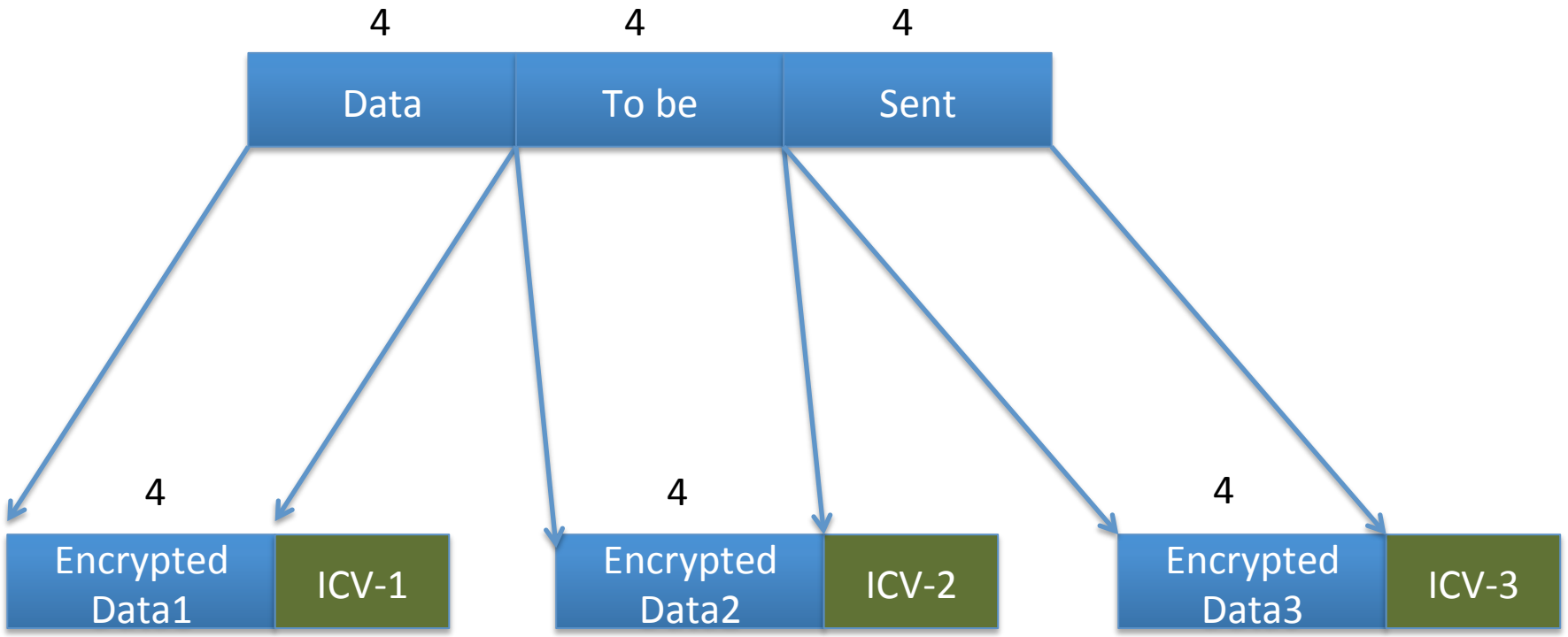


Figure 9-2—Fragmentation



Fragmentation to the Rescue



- Up to 16 fragments can be sent
- Each can carry 4 bytes of data
- Total 64 bytes can be injected

Hirte Attack

- Uses key concepts from the Caffe Latte attack and Fragmentation attack
- Targets an isolated client, allows association, waits for an ARP packet like the Caffe Latte
- Converts that into an ARP Request for the same client by relocating the IP address in the ARP header using fragmentation and patches ICV using Message Modification flaw
- Client accepts packet and sends replies
- GAME OVER!

More details

The Final Nail in WEP's Coffin

Andrea Bittau
University College London
a.bittau@cs.ucl.ac.uk

Mark Handley
University College London
m.handley@cs.ucl.ac.uk

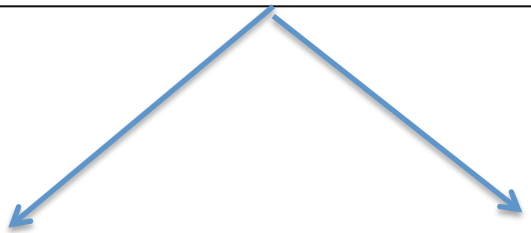
Joshua Lackey
Microsoft
joshlack@microsoft.com

- Paper detailing fragmentation and its advanced use
- Aircrack-ng website for details on implementation

We need WEP's Replacement

WPA

- Intermediate solution by Wi-Fi Alliance
- Uses TKIP
  - Based on WEP
- Hardware changes not required
- Firmware update



Personal

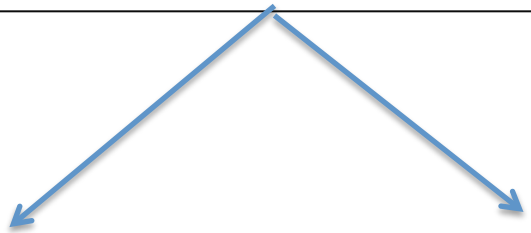
PSK

Enterprise

802.1x + Radius

WPA2

- Long Term solution (802.11i)
- Uses CCMP
  - Based on AES
- Hardware changes required



Personal

PSK

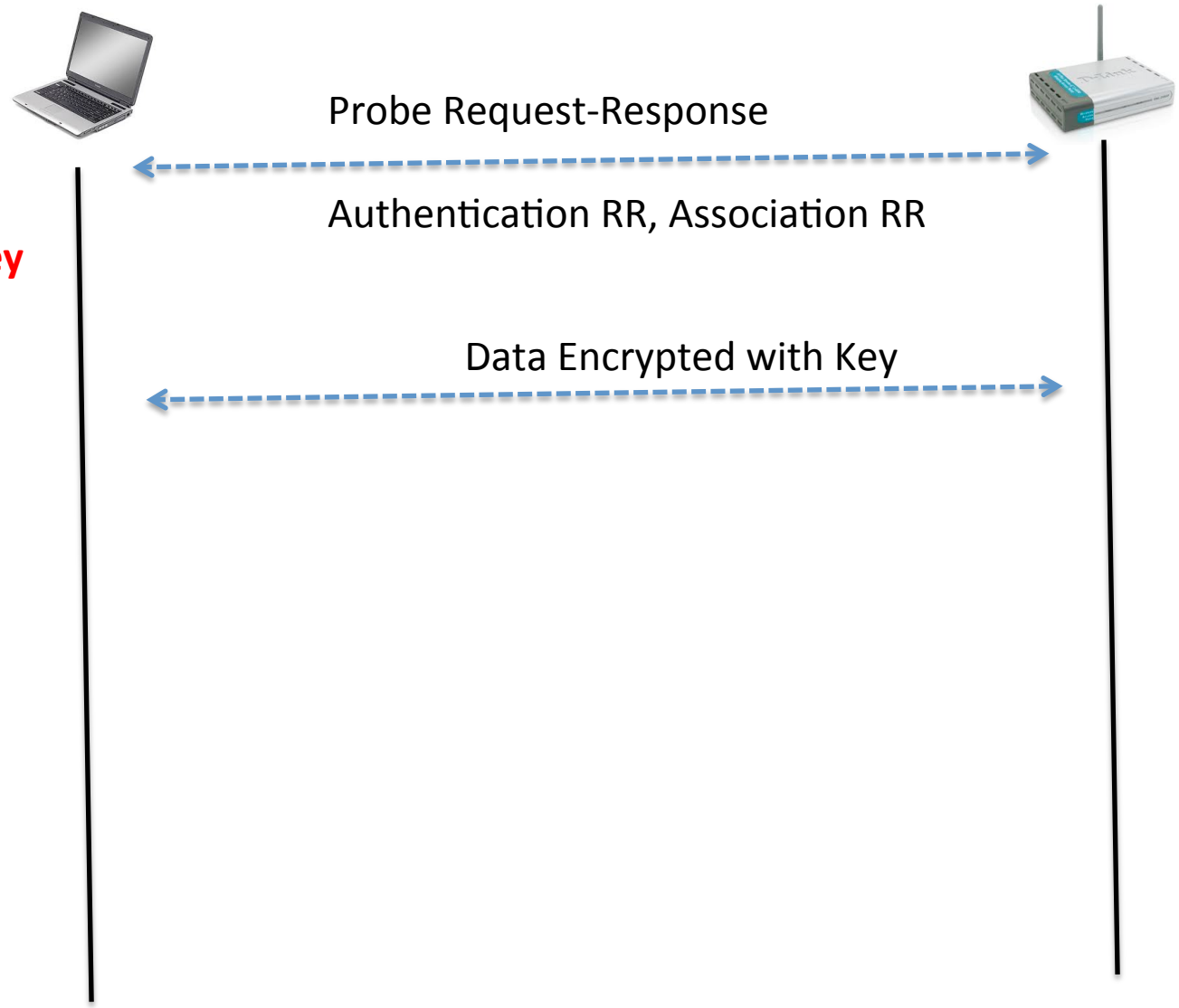
Enterprise

802.1x + Radius

How does the Client Know?

- Beacon Frames?
- Probe Response Packets from the AP?
- Can be used to create a WPA/WPA2 Honeypot as well!

WEP



WPA: No Static Keys

Probe Request-Response

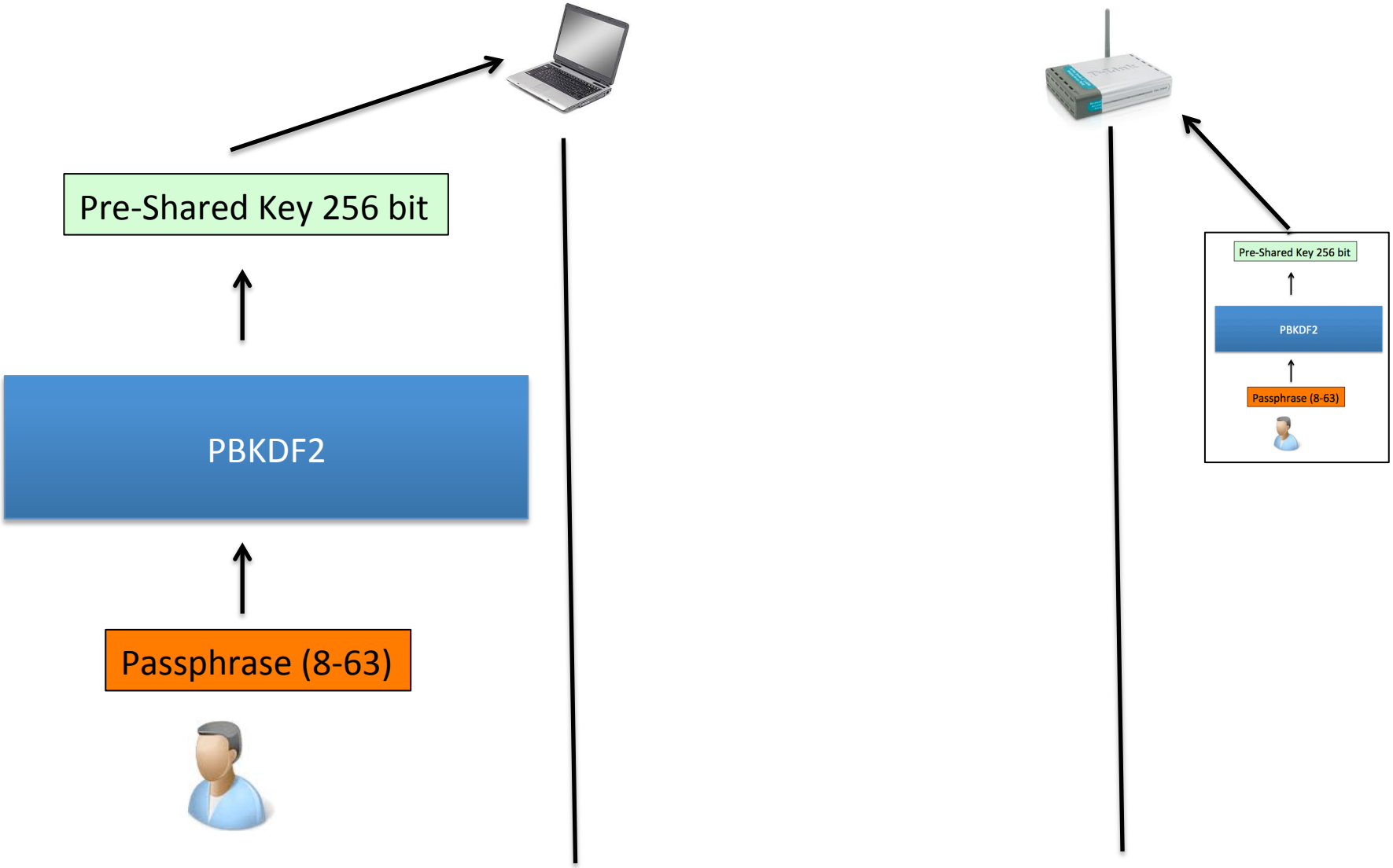
Authentication RR, Association RR

Dynamic Key Generated First

Data Encrypted with Dynamically Key

How are Dynamic Keys Created?

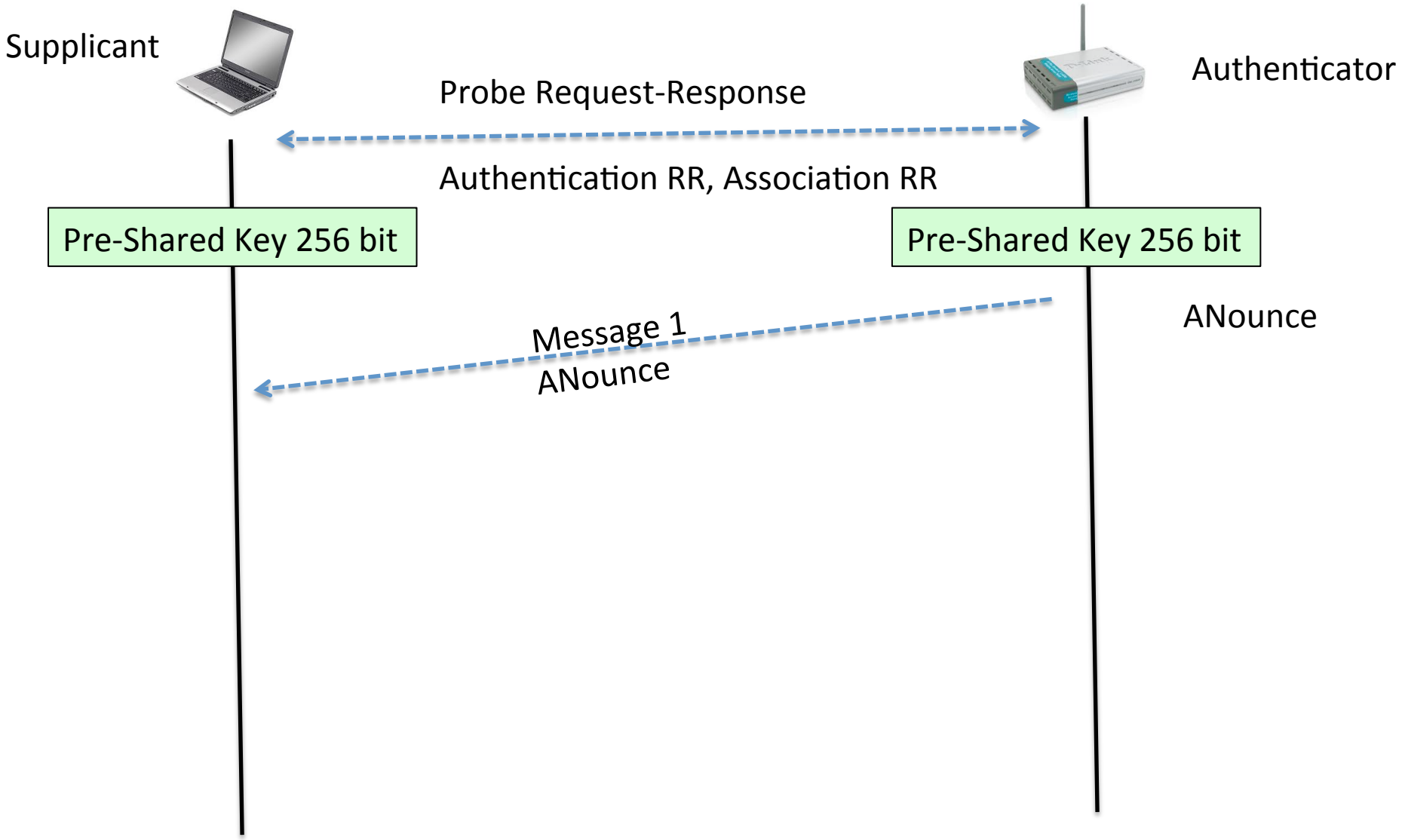
WPA Pre-Shared Key



PBKDF2

- Password Based Key Derivation Function
- RFC 2898
- PBKDF2(Passphrase, SSID, ssidLen, 4096, 256)
- 4096 – Number of times the passphrase is hashed
- 256 – Intended Key Length of PSK

Lets “Shake Hands”: 4-Way Handshake



Message 1

8.5.3.1 4-Way Handshake Message 1

Message 1 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)

Key Type = 1 (Pairwise)

Install = 0

Key Ack = 1

Key MIC = 0

Secure = 0

Error = 0

Request = 0

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite-specific; see Table 20f

Key Replay Counter = *n* – to allow Authenticator to match the right Message 2 from Supplicant

Key Nonce = ANonce

EAPOL-Key IV = 0

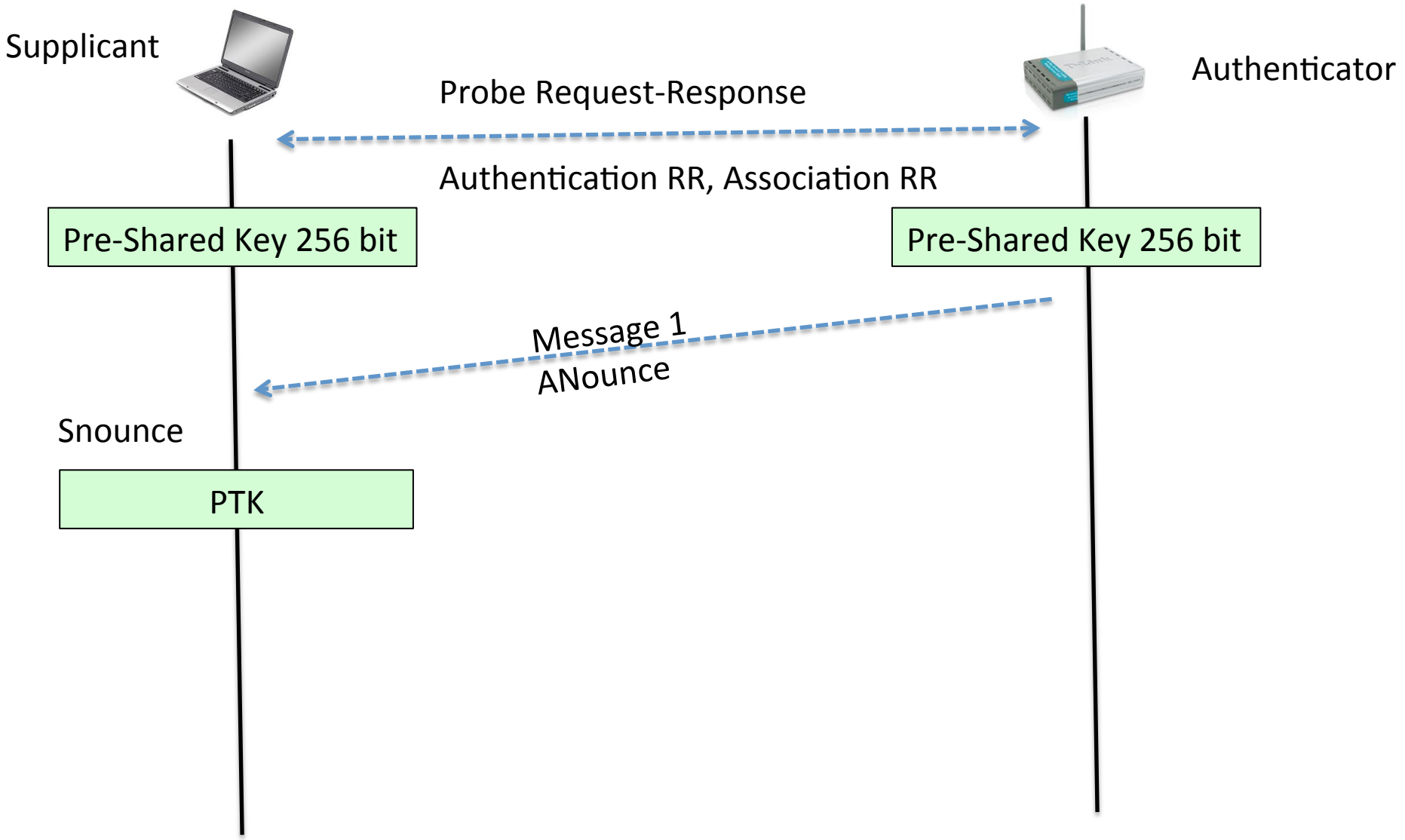
Key RSC = 0

Key MIC = 0

Key Data Length = 22

Key Data = PMKID for the PMK being used during this exchange

4 Way Handshake: Message 1



Pairwise Transient Key

PTK = Function(PMK, ANounce, SNounce, Authenticator MAC, Supplicant MAC)

-PMK = Pre-Shared Key (Pairwise Master Key)

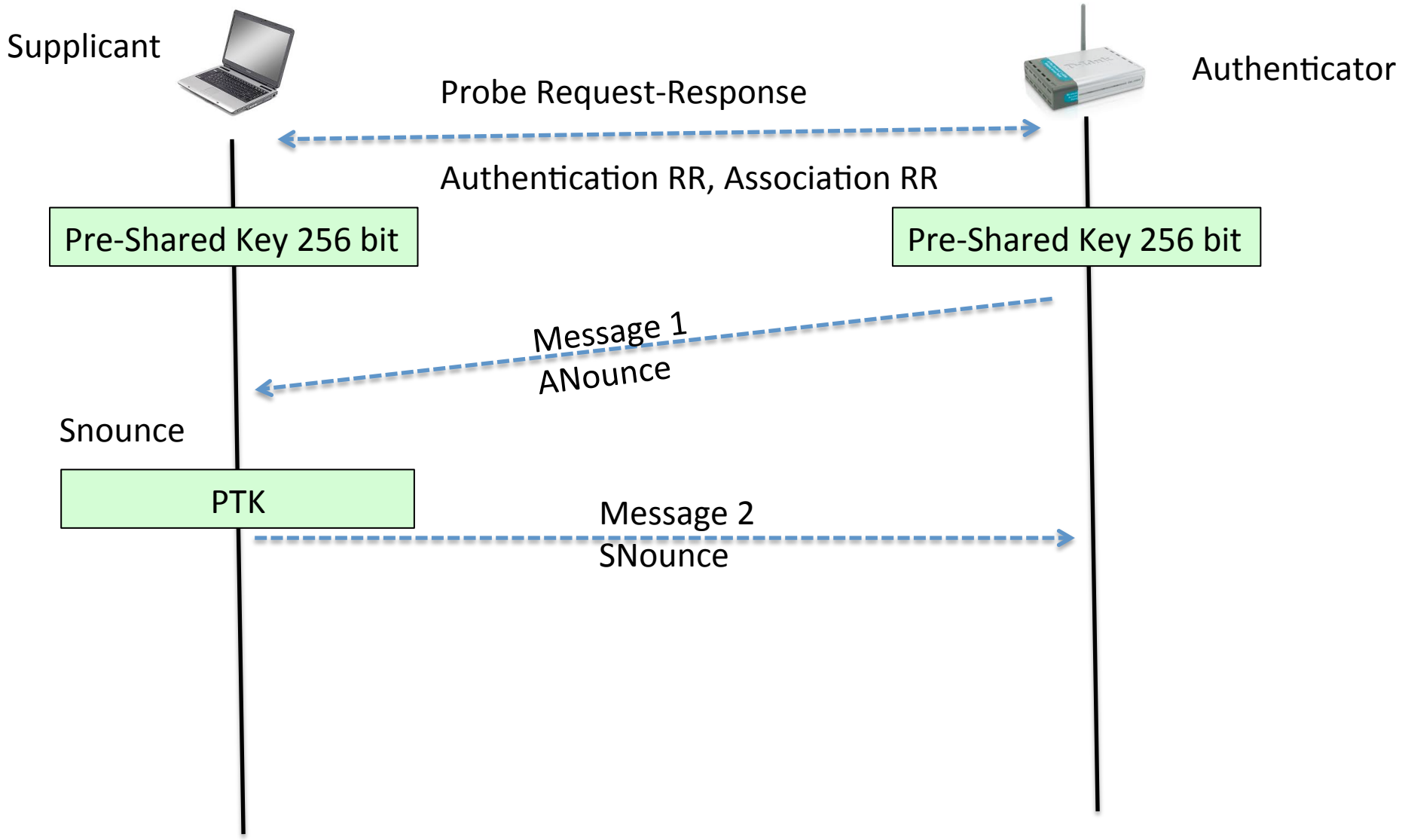
-ANounce = Random by AP

-SNounce = Random by Client

-Authentication MAC = AP MAC

-Supplicant MAC = Client MAC

4 Way Handshake: Message 2



Message 2

8.5.3.2 4-Way Handshake Message 2

Message 2 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

Install = 0

Key Ack = 0

Key MIC = 1

Secure = 0 – same as Message 1

Error = 0 – same as Message 1

Request = 0 – same as Message 1

Encrypted Key Data = 0

Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n* – to let the Authenticator know to which Message 1 this corresponds

Key Nonce = SNonce

EAPOL-Key IV = 0

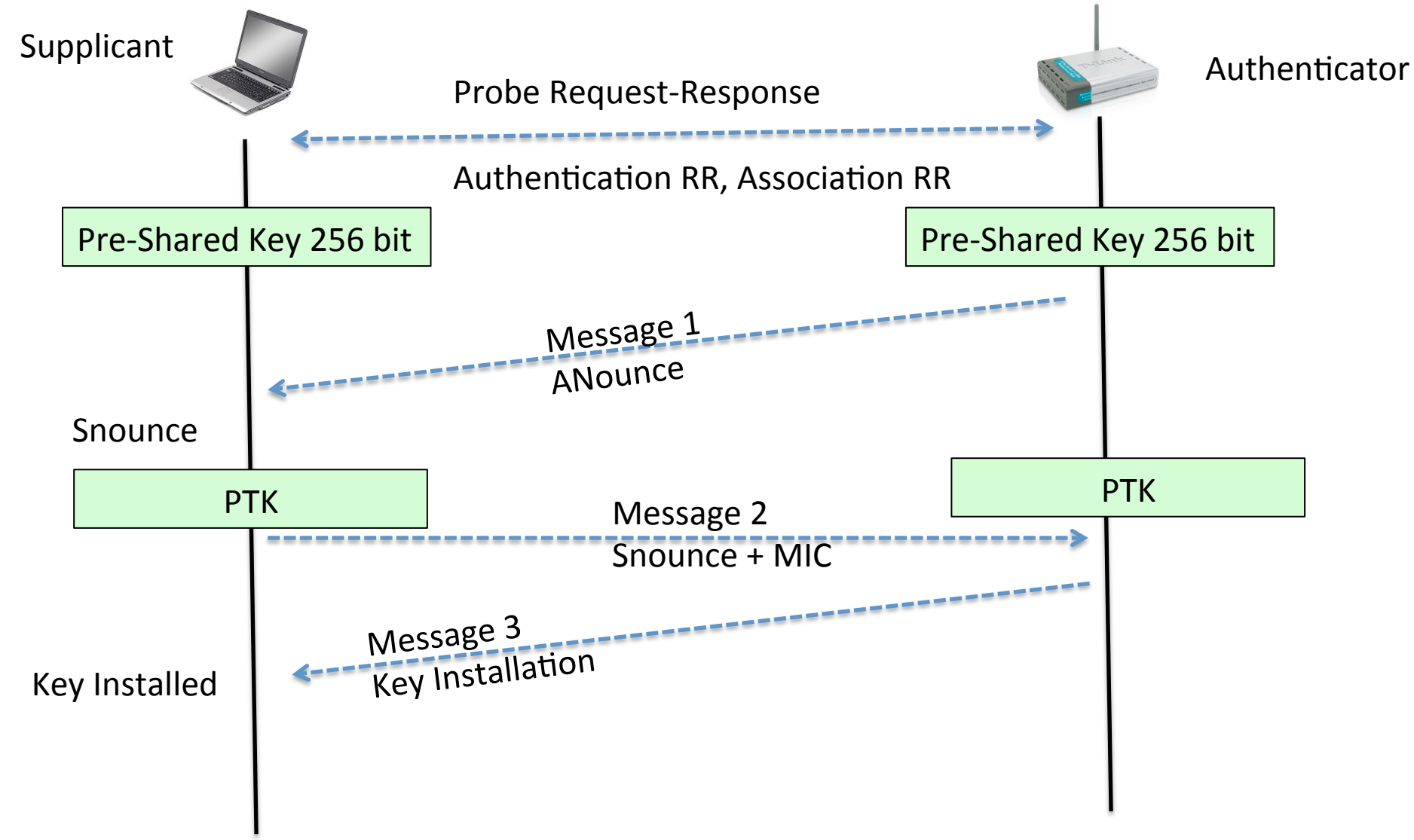
Key RSC = 0

Key MIC = MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length in octets of included RSN information element

Key Data = included RSN information element – the sending STA's RSN information element

4 Way Handshake: Message 3



Message 3

8.5.3.3 4-Way Handshake Message 3

Message 3 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1

Key Type = 1 (Pairwise) – same as Message 1

Install = 0/1 – 0 only if the AP does not support key mapping keys, or if the STA has the No Pairwise bit (in the RSN Capabilities field) set and only the group key will be used

Key Ack = 1

Key MIC = 1

Secure = 1 (keys installed)

Error = 0 – same as Message 1

Request = 0 – same as Message 1

Encrypted Key Data = 1

Reserved = 0 – unused by this protocol version

Key Length = Cipher-suite-specific; see Table 20f

Key Replay Counter = *n*+*1*

Key Nonce = ANonce – same as Message 1

EAPOL-Key IV = 0 (Version 2) or random (Version 1)

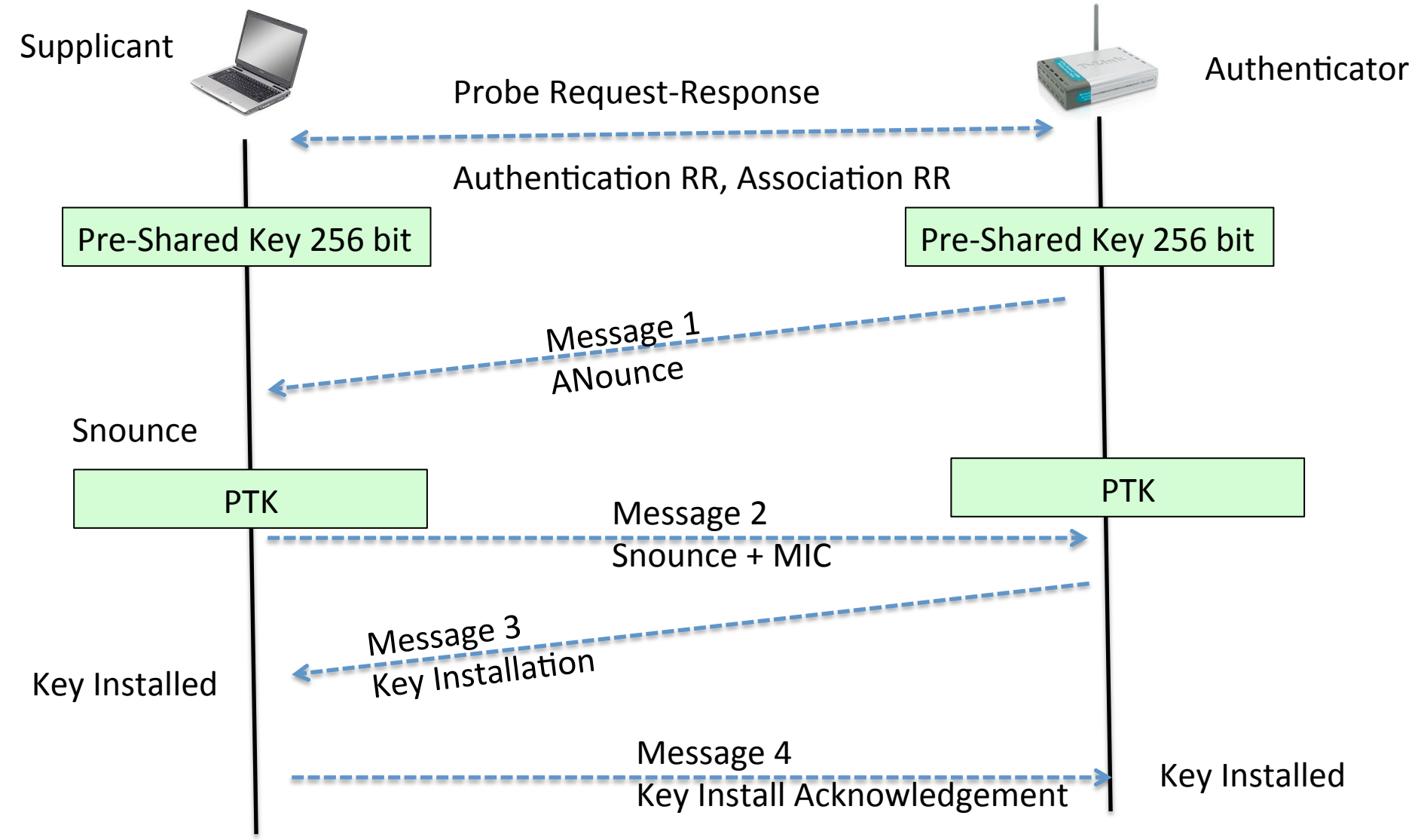
Key RSC = starting sequence number that the Authenticator's STA will use in MPDUs protected by GTK

Key MIC = MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

Key Data Length = length in octets of included RSN information elements and GTK

Key Data = the AP's Beacon/Probe Response frame's RSN information element, and, optionally, a second RSN information element that is the Authenticator's pairwise cipher suite assignment, and, if a group cipher has been negotiated, the encapsulated GTK and the GTK's key identifier (see 8.5.2)

4 Way Handshake: Message 4



Message 4

8.5.3.4 4-Way Handshake Message 4

Message 4 uses the following values for each of the EAPOL-Key frame fields:

Descriptor Type = N – see 8.5.2

Key Information:

- Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128) – same as Message 1
- Key Type = 1 (Pairwise) – same as Message 1
- Install = 0
- Key Ack = 0 – this is the last message
- Key MIC = 1
- Secure = 1
- Error = 0
- Request = 0
- Encrypted Key Data = 0
- Reserved = 0 – unused by this protocol version

Key Length = 0

Key Replay Counter = *n+1*

Key Nonce = 0

EAPOL-Key IV = 0

Key RSC = 0

Key MIC = MIC(KCK, EAPOL) – MIC computed over the body of this EAPOL-Key frame with the Key MIC field first initialized to 0

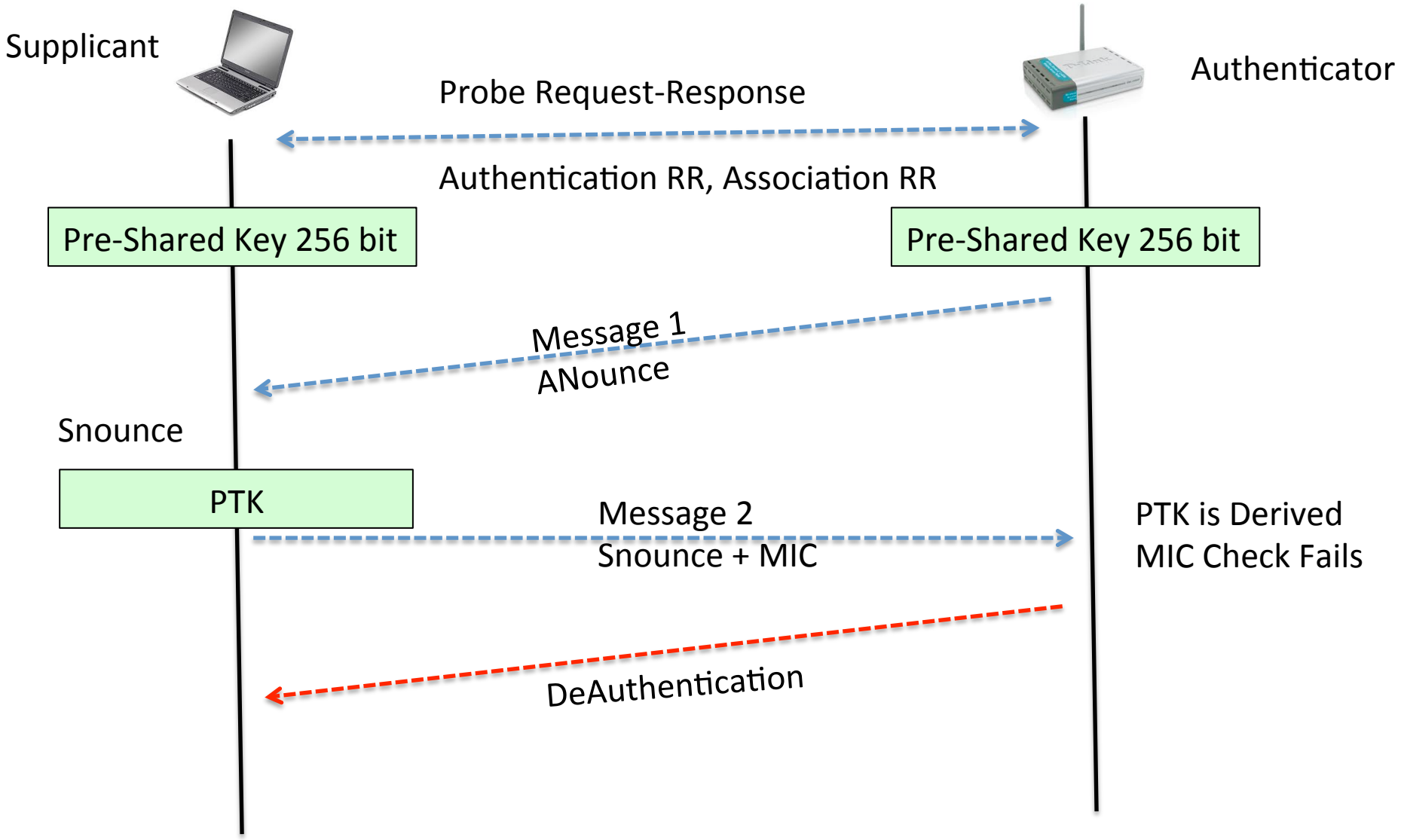
Key Data Length = 0

Key Data = none required

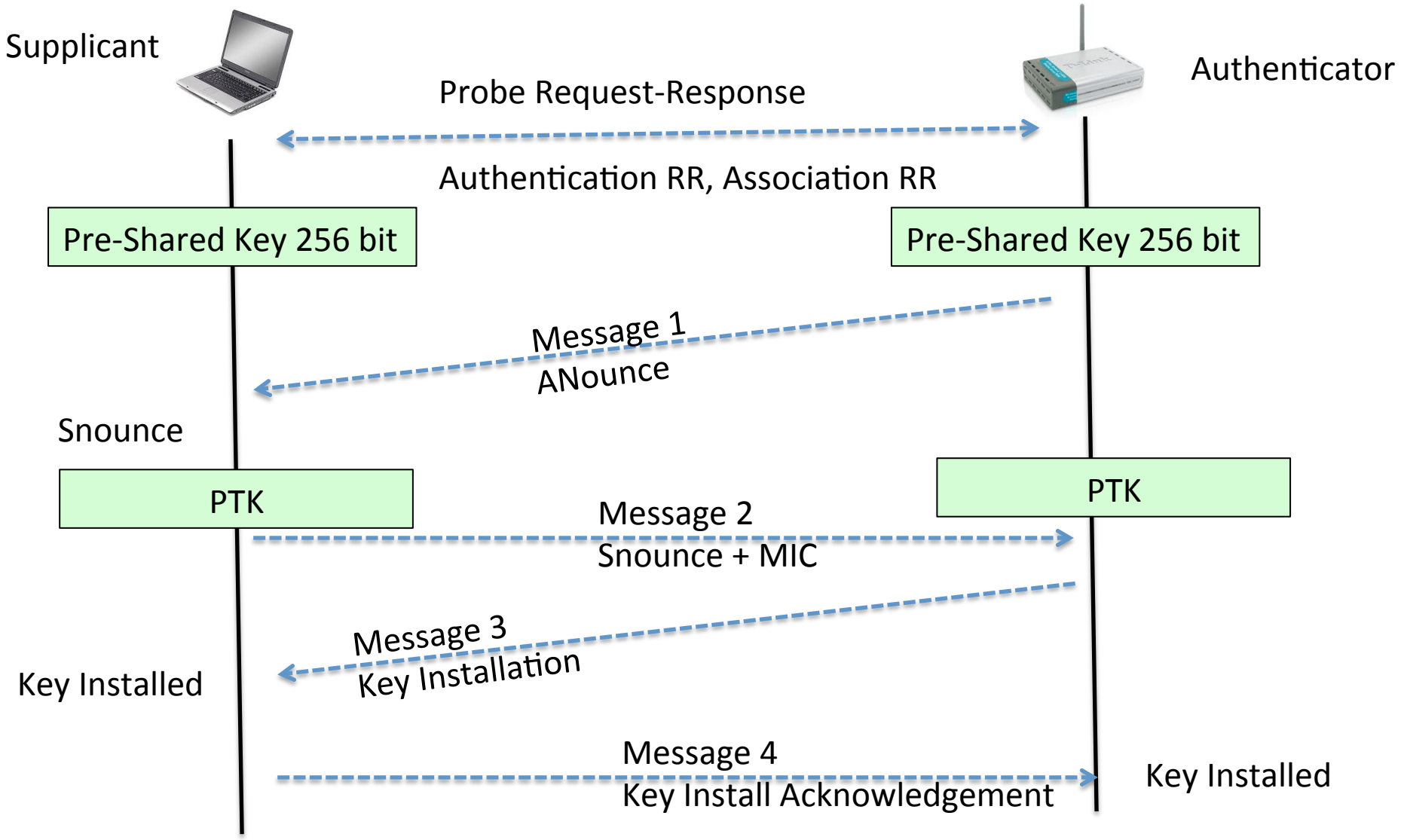
Acknowledgements

- IEEE Standard 802.11i-2004

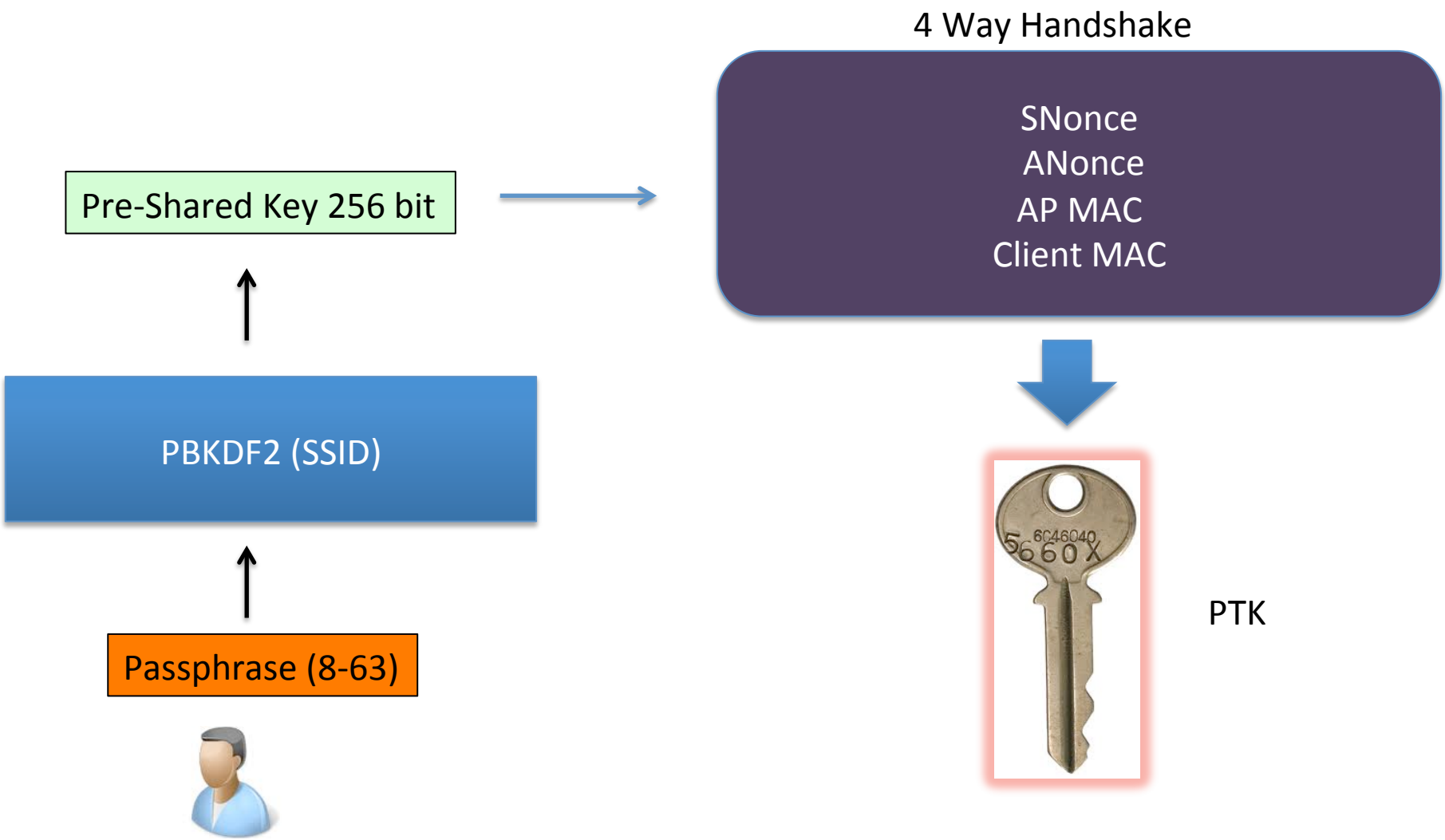
Dunno the Right Phrase?



Eavesdropping the 4 Way Handshake



A Quick Block Diagram



WPA-PSK Dictionary Attack

4 Way Handshake

SNonce
ANonce
AP MAC
Client MAC

Pre-Shared Key 256 bit

PBKDF2 (SSID)

Passphrase (8-63)

Dictionary

PTK

Verify by Checking the MIC

Which Packet Do we Need in the Handshake?

- All Packets have the AP MAC and Client MAC
- ANonce
  - Packet 1 and Packet 3
- SNonce
  - Packet 2

Answer: (Either All 4 packets), or (packet 1 and 2) or (packet 2 and 3)

Decrypting WPA-PSK Traces

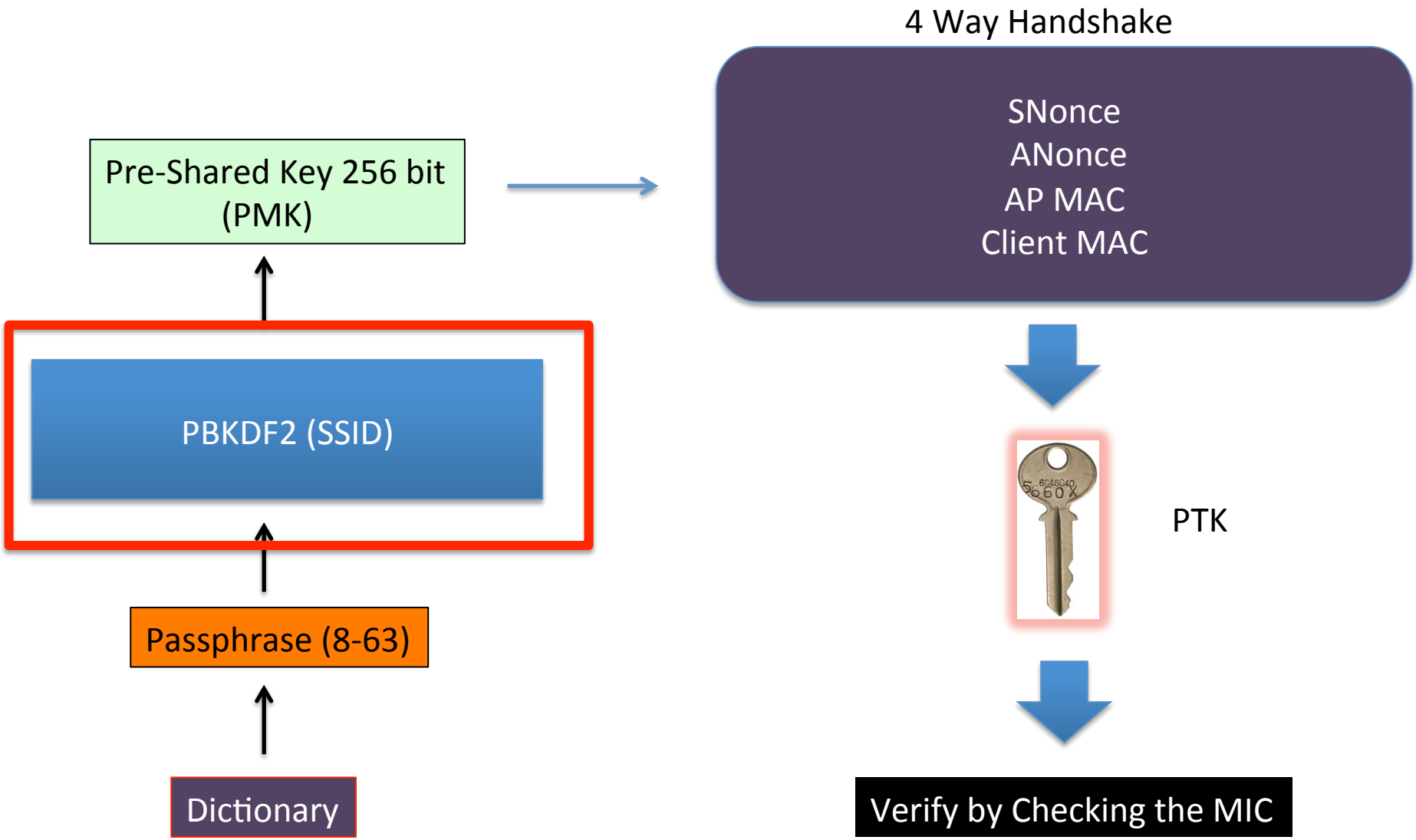
- Wireshark
- Airdecap-NG

Cracking WPA2-PSK

- Same principles apply
- As vulnerable as WPA-PSK is if a weak passphrase is chosen
- Nothing extra to discuss

Demo Time!

WPA-PSK Dictionary Attack



PBKDF2

- Requires SSID
  - List of commonly used SSIDs
- Requires Passphrase
  - Can be provided from a Dictionary

- PMK can be pre-computed using the above

Other Parameters in Key Cracking

- Snonce, Anonce, Supplicant MAC, Authenticator MAC varies and hence cannot be “pre-calculated”
- PTK will be different based on the above
- MIC will be different as well

Thus these cannot be pre-calculated in any way

Speeding up Cracking

4 Way Handshake

Pre-Shared Key 256 bit (PMK)

SNonce
ANonce
AP MAC
Client MAC

Pre-Calculated List of PMK for a
1. Given SSID
2. Dictionary of Passphrases

PTK

Verify by Checking the MIC

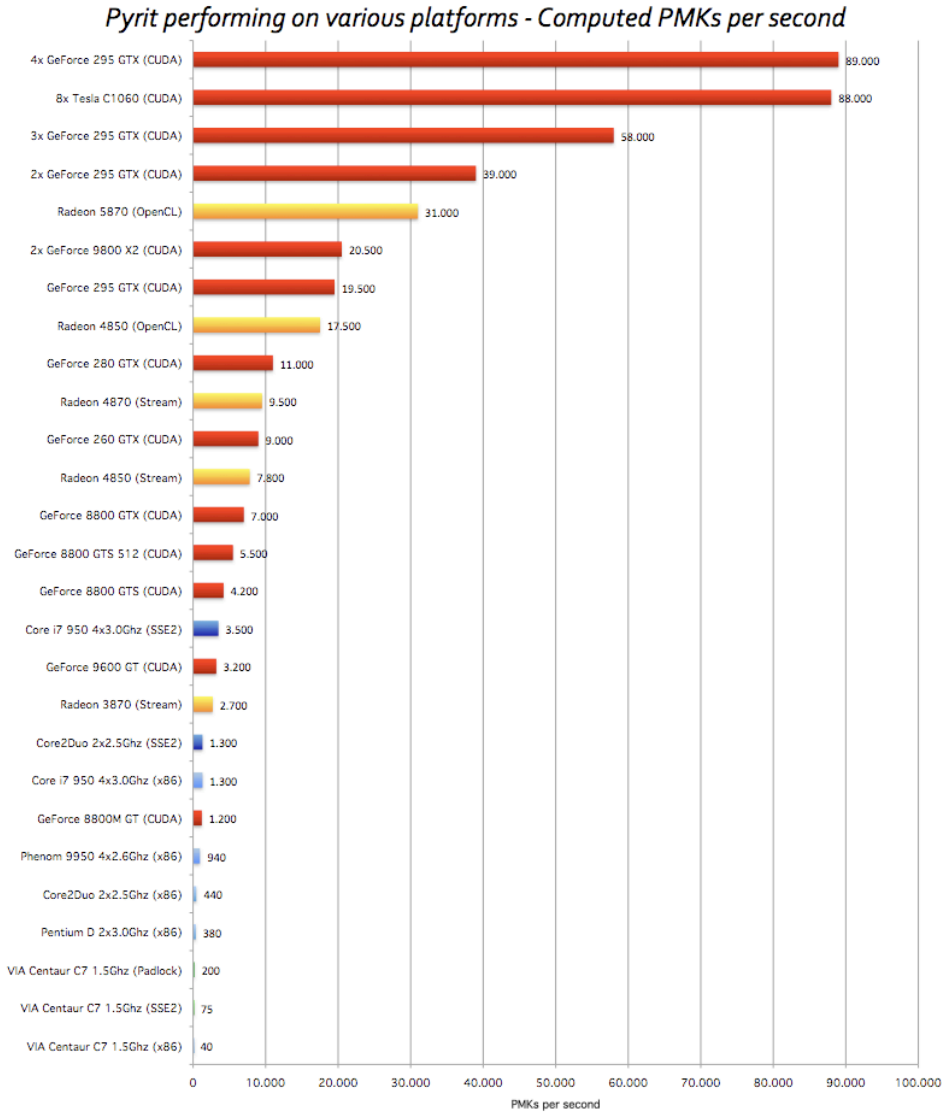
Platforms

- Multi-Cores
- ATI-Stream
- Nvidia CUDA
- ….
- In the Cloud
  - Amazon EC2

Fast Cracking Demo

- Pyrit

  http://code.google.com/p/pyrit/



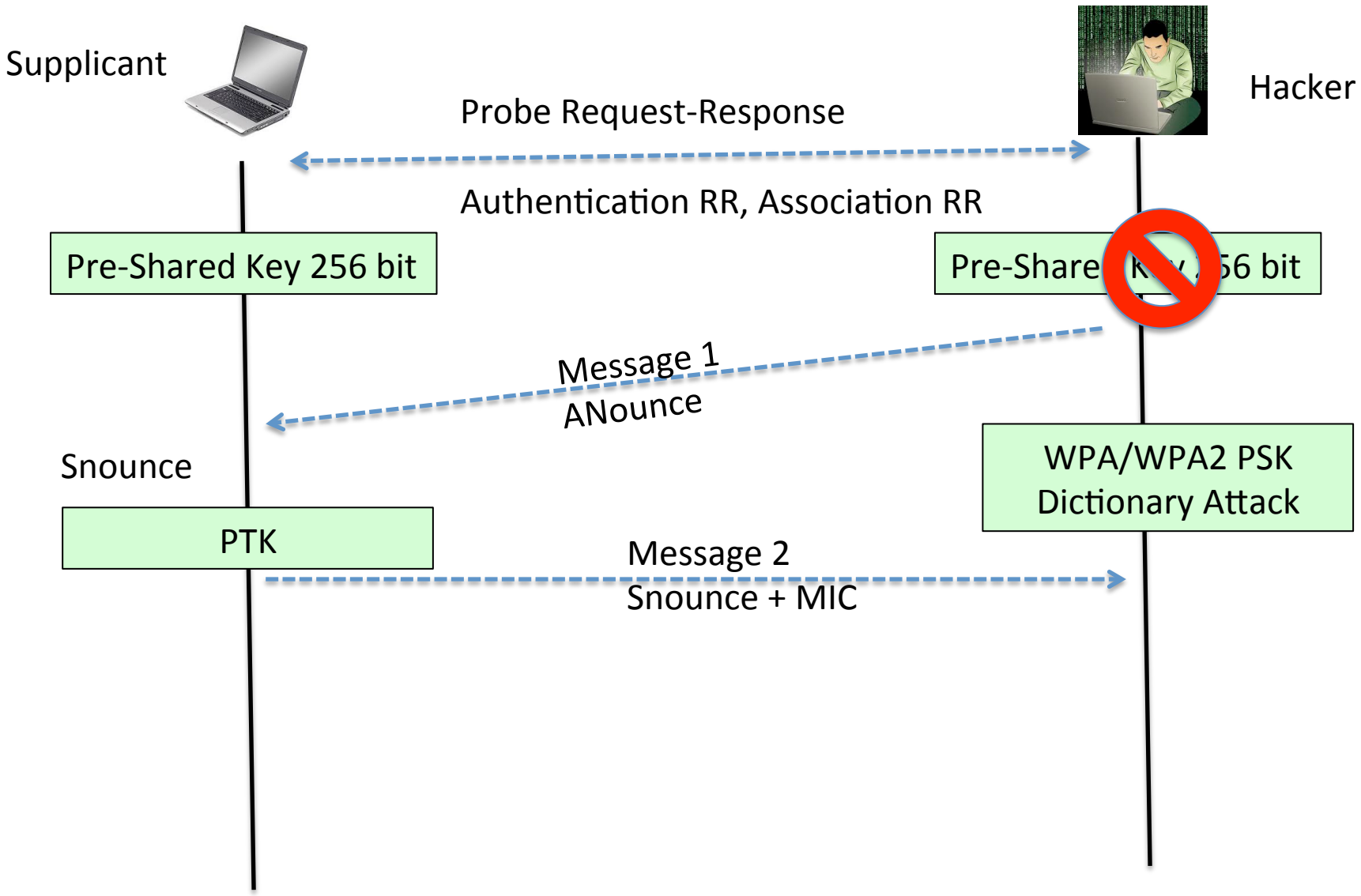
Stories of a Wandering Client

- Multiple Profiles stored
  - Open
  - WEP
  - WPA/WPA2
- Tools don't work properly (WiFish Finder etc.)
- But lets crack this from the basic principles

Exploit All Possibilities

- Need SSID with multiple configurations
- We need to find the security settings first
- We will fight the battle later

Stimulating a Handshake



Connecting to WPA/WPA2 Networks

- WPA_Supplicant is the de-facto tool
- Supports tons of options
- Cross Platform
  - Linux
  - Windows
  - OS X
- Allows for better understanding of process
- Open source

Supported EAP Methods

Supported EAP methods (IEEE 802.1X Supplicant)

- EAP-TLS
- EAP-PEAP/MSCHAPv2 (both PEAPv0 and PEAPv1)
- EAP-PEAP/TLS (both PEAPv0 and PEAPv1)
- EAP-PEAP/GTC (both PEAPv0 and PEAPv1)
- EAP-PEAP/OTP (both PEAPv0 and PEAPv1)
- EAP-PEAP/MD5-Challenge (both PEAPv0 and PEAPv1)
- EAP-TTLS/EAP-MD5-Challenge
- EAP-TTLS/EAP-GTC
- EAP-TTLS/EAP-OTP
- EAP-TTLS/EAP-MSCHAPv2
- EAP-TTLS/EAP-TLS
- EAP-TTLS/MSCHAPv2
- EAP-TTLS/MSCHAP
- EAP-TTLS/PAP
- EAP-TTLS/CHAP
- EAP-SIM
- EAP-AKA
- EAP-AKA'
- EAP-PSK
- EAP-FAST
- EAP-PAX
- EAP-SAKE
- EAP-IKEv2
- EAP-GPSK
- LEAP (note: requires special support from the driver)

Configuration File Required

- Samples available on the tool website
- Best idea is to use available templates and customize

WPA-Enterprise

- Use a RADIUS server for authentication
- Different supported EAP types – EAP-MD5, PEAP, EAP-TLS etc.
- De facto server
  - FreeRadius www.freeradius.org
- Depending on EAP type used Client and Server will need to be configured

FreeRadius-WPE

- FreeRadius Wireless Pwnage Edition ☺
- Created by Joshua and Brad
- A patch to the FreeRadius code http://www.willhackforsushi.com/?page_id=37

Key Benefits (ripped from Josh's site)

- Simplifies the setup of FreeRADIUS by adding all RFC1918 addresses as acceptable NAS devices;
- Simplifies the setup of EAP authentication by including support for all FreeRADIUS supported EAP types;
- Adds WPE logging in $prefix/var/log/radius/freeradius-server-wpe.log, can be controlled in radius.conf by changing the "wpelogfile" directive;
- Simplified the setup of user authentication with a default "users" file that accepts authentication for any username;
- Adds credential logging for multiple EAP types including PEAP, TTLS, LEAP, EAP-MD5, EAP-MSCHAPv2, PAP, CHAP and others

Good news – BT5 ships with FreeRadius-WPE
Bad News – Broken by default

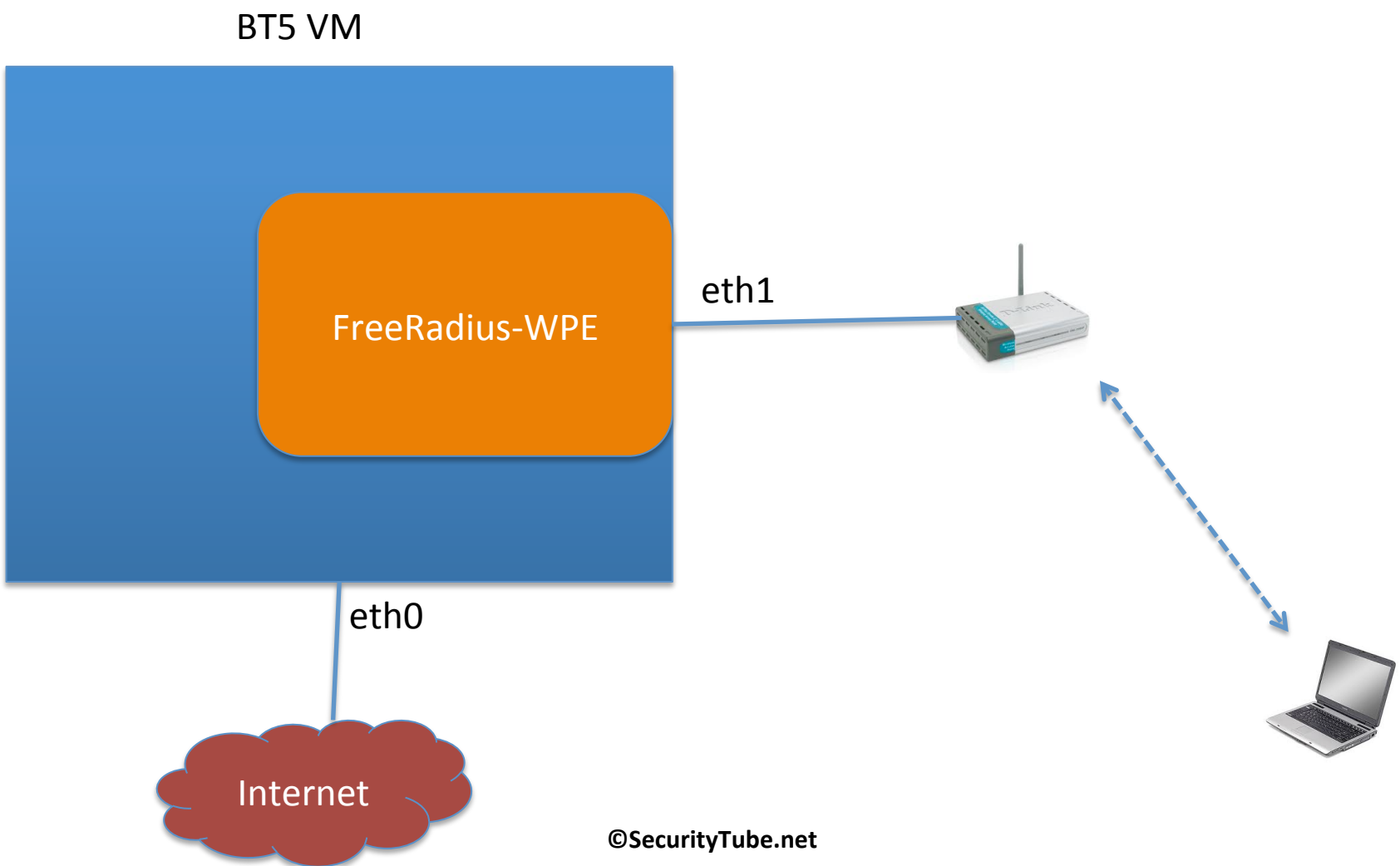
Good news – Easy fix

http://redmine.backtrack-linux.org:8080/issues/115

Setting up FreeRadius-WPE

- Fixing problems on BT5
- Recompilation
- Basic usage

Network Architecture



EAP-MD5

Client

AP

RADIUS

EAP Request Identity

EAP Response Identity

RADIUS Access Request
(EAP Identity Response)

EAP Request MD5 Challenge

RADIUS Access Challenge
(EAP Request MD5 Challenge)

EAP Response MD5 Challenge

RADIUS Access Request
(EAP Response MD5 Challenge)

EAP Success

RADIUS Access Accept
(EAP Success)

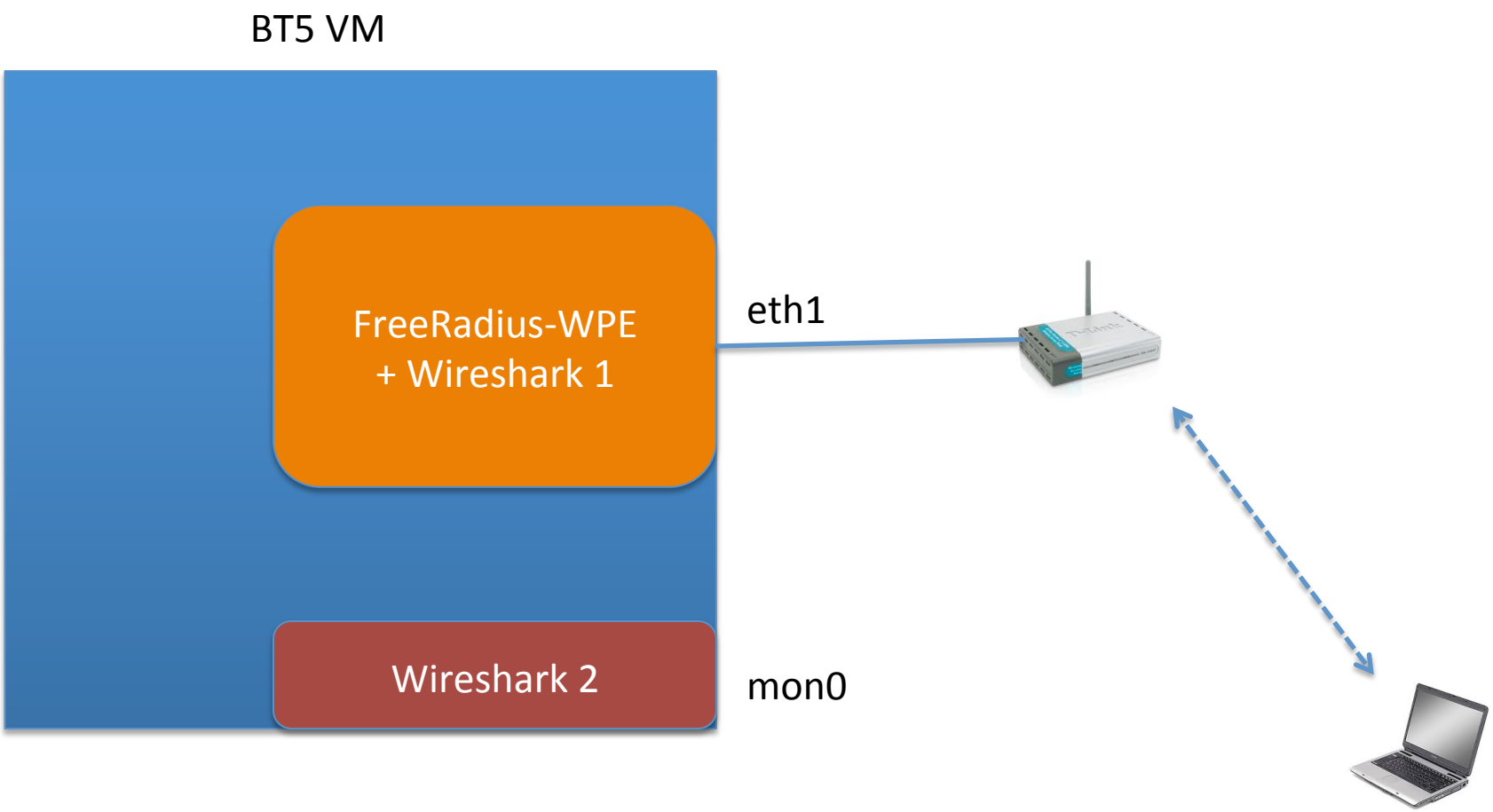
Setting up the RADIUS Server

- Add a username / password in users file
- Make eap-md5 the default EAP type in eap.conf
- Ensure the shared secret is correct for the AP-RADIUS server in clients.conf

Objective of Lab

- Observe traffic on wired side between AP and RADIUS
- Observer traffic on wireless side between Client and AP
- Understand and correlate with the theory

Network Architecture



EAP-MD5

- Cannot be used for Wi-Fi as does not support key generation
- Does not support mutual authentication
- Both plaintext challenge and response goes over the air unencrypted
  - Attacker can obtain both
  - Launch a dictionary / educated bruteforce attack

MD5 Mathemagic

Hash = MD5(EAP Response ID + Password + RADIUS Challenge)

Available to attacker:

- Hash
- Response ID
- Challenge

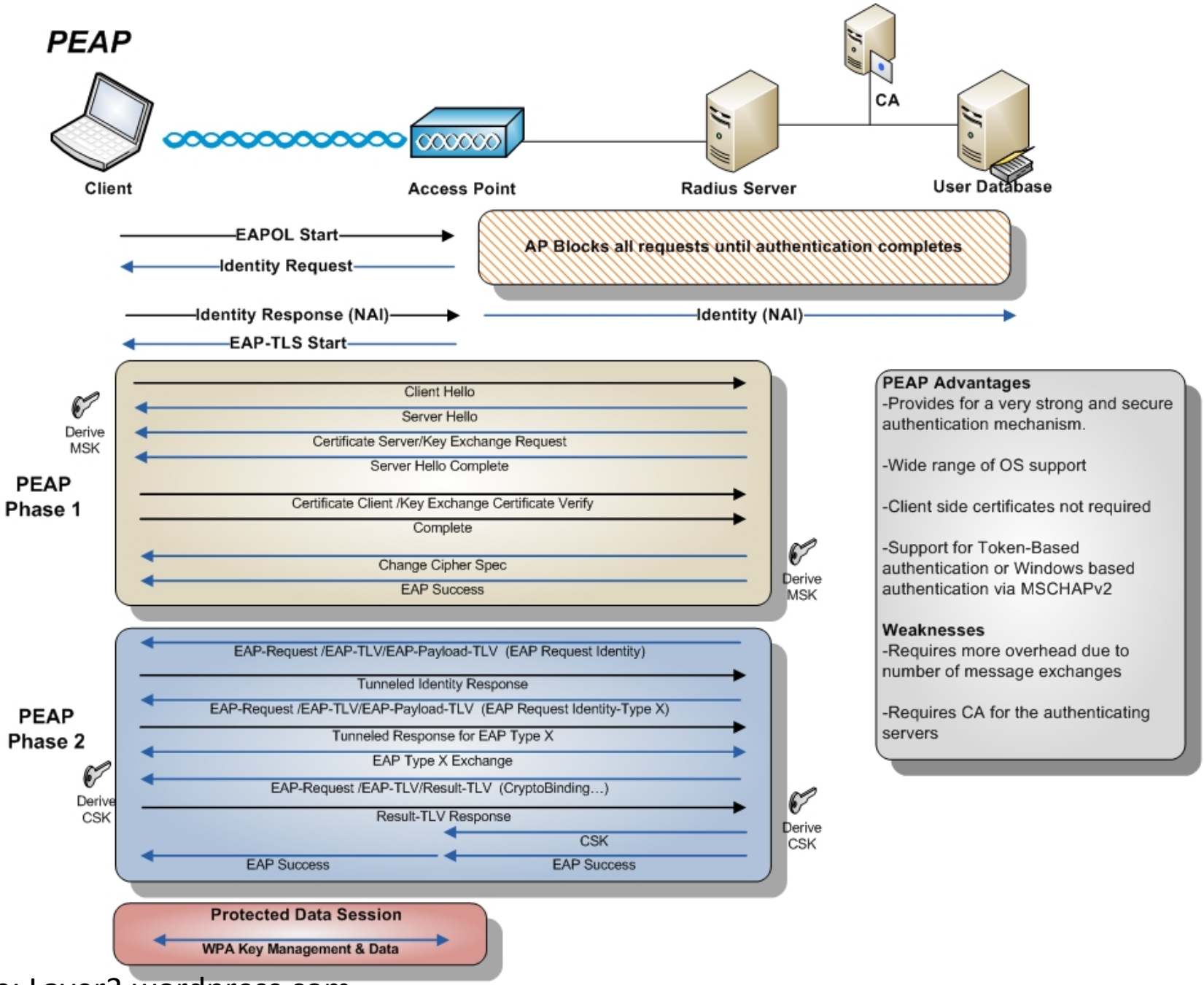
- Simple equation
  - Keep guessing password till the Hash matches

WPA/WPA2 Enterprise

EAP Type	Real World Usage
PEAP	Highest
EAP-TTLS	High
EAP-TLS	Medium
LEAP	Low
EAP-FAST	Low
….	….

PEAP

- Protected Extensible Authentication Protocol
- Typical usage:
  - PEAPv0 with EAP-MSCHAPv2 (most popular)
    - Native support on Windows
  - PEAPv1 with EAP-GTC
- Other uncommon ones
  - PEAPv0/v1 with EAP-SIM (Cisco)
- Uses Server Side Certificates for validation
- PEAP-EAP-TLS
  - Additionally uses Client side Certificates or Smartcards
  - Supported only by Microsoft



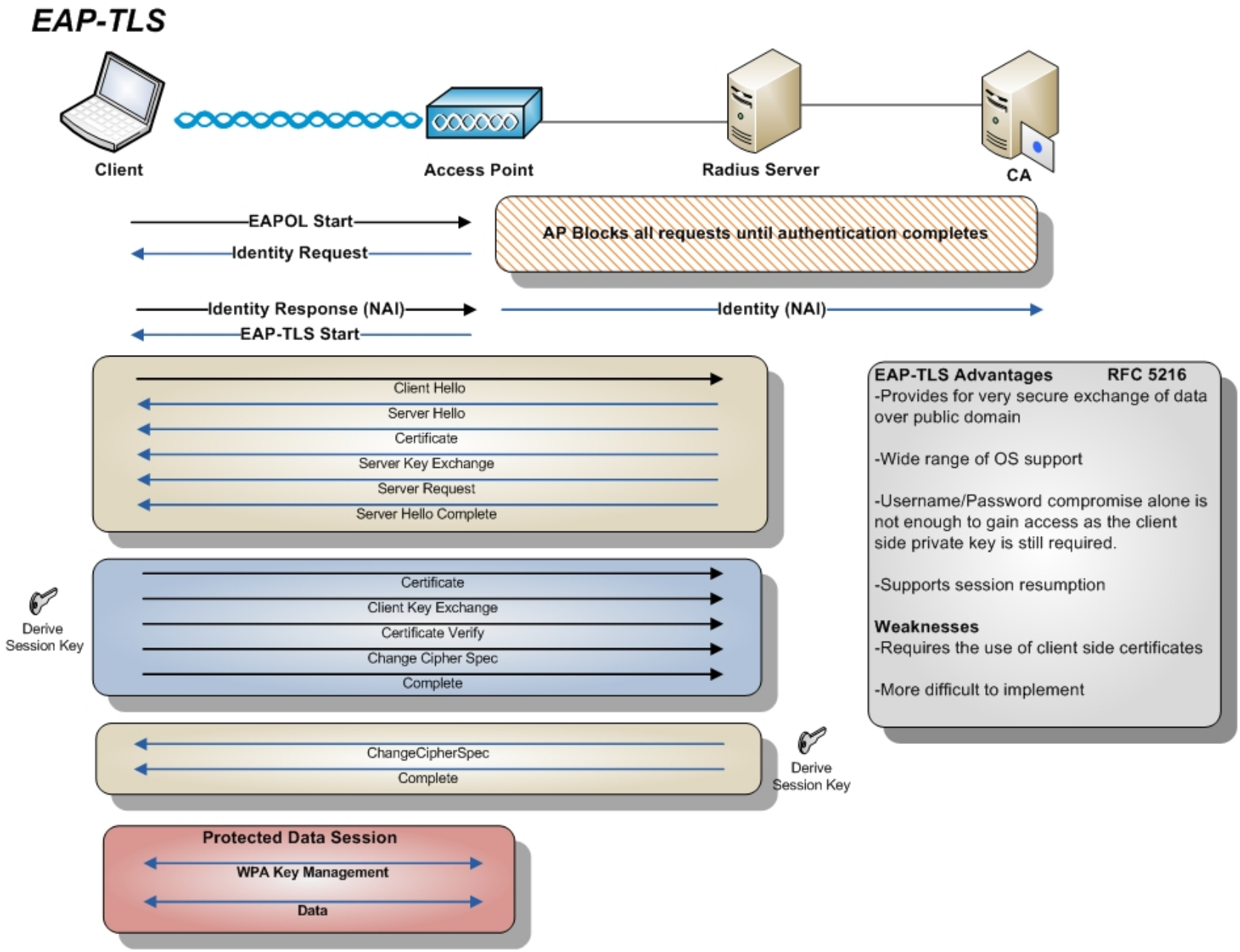
Source: Layer3.wordpress.com

EAP-TTLS

- EAP-Tunneled Transport Layer Security
- Server authenticates with Certificate
- Client can optionally use Certificate as well
- No native support on Windows
  - 3rd party utilities to be used
- Versions
  - EAP-TTLSv0
  - EAP-TTLSv1

EAP-TLS

- Strongest security of all the EAPs out there
- Mandates use of both Server and Client side certificates
- Required to be supported to get a WPA/WPA2 logo on product
- Unfortunately, this is not very popular due to deployment challenges

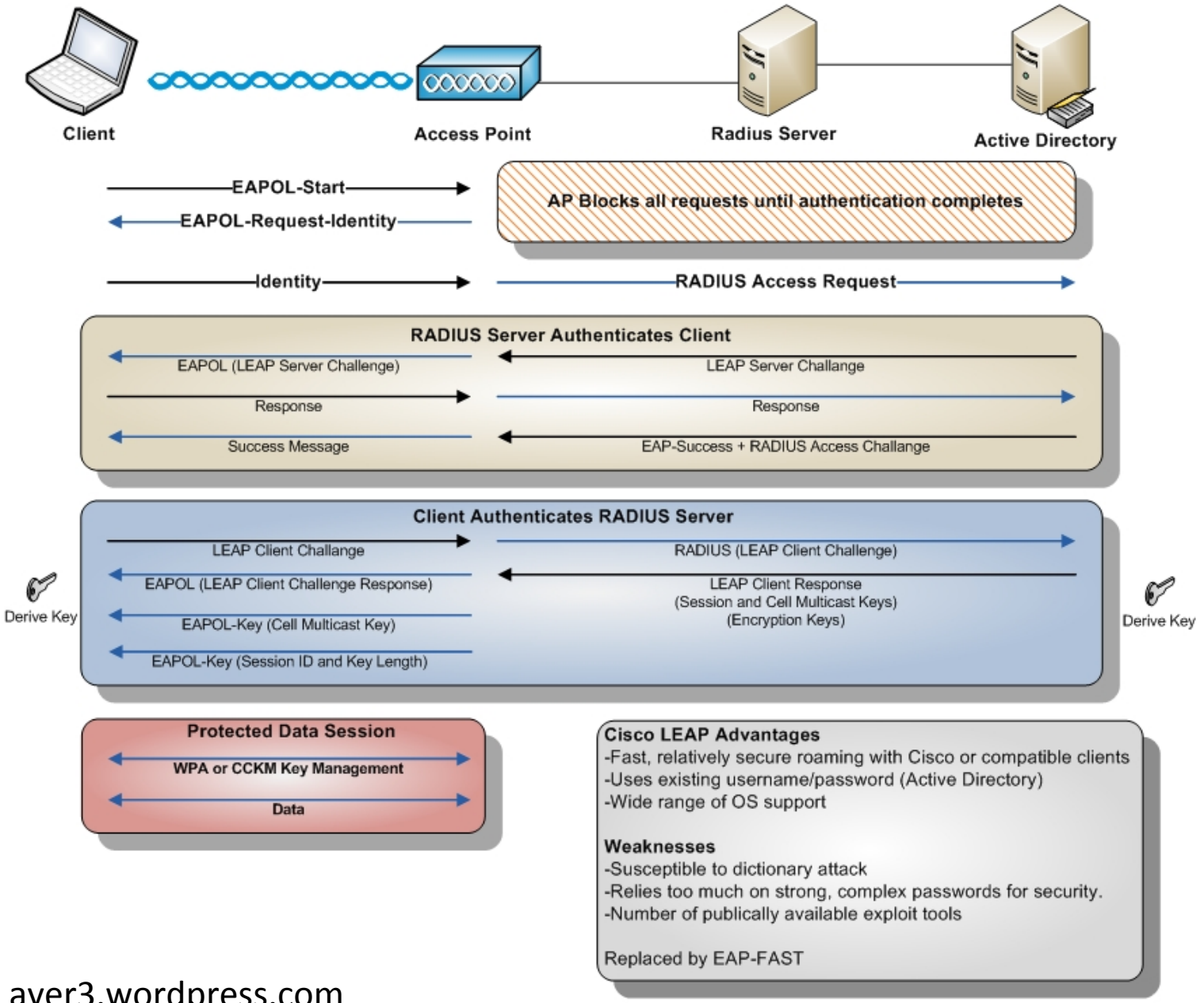


Source: Layer3.wordpress.com

LEAP

- Lightweight Extensible Authentication Protocol
- Cisco proprietary
- Almost obsolete now due to security issues – dictionary attacks
  - Modified MSCHAP for authentication
- Cisco recommends usage of EAP-FAST as the official LEAP replacement

Cisco LEAP

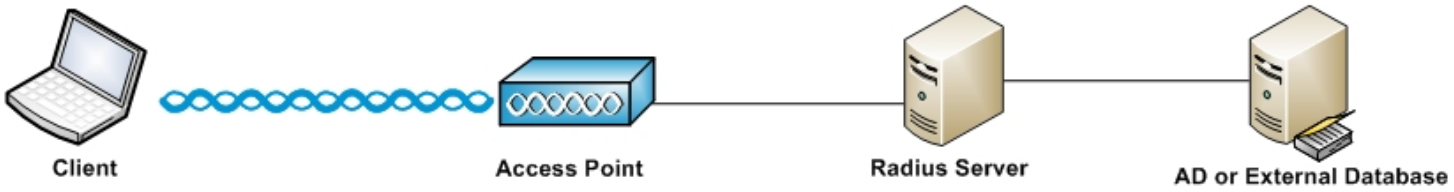


Source: Layer3.wordpress.com

EAP-FAST

- Touted by Cisco as a replacement for LEAP
- Server side certificates are optional
- Uses Protected Access Credential (PAC) to create a TLS tunnel
  - Provisioning can be manual / automated per user
- Credentials are exchanged over the TLS tunnel
- Automatic PAC provisioning is susceptible to interception attack
- A Honeypot AP can be used by the attacker to update the PAC and then bruteforce the MSCHAPv2 credentials supplied by the client

EAP-FAST



Start

EAP Identity Request

AP Blocks all requests until authentication completes

EAP Identity Response

Identity

Server Side Authentication - Establish a Secure Tunnel (PAC and TLS)

EAP-FAST Start (A-ID)

A-ID

EAP-FAST (TLS Client Hello) PAC-OPAQUE

PAC-Opaque

Server_random

EAP-FAST (TLS Server Hello) Server_random

TLS Finished

Client Side Authentication – Server Authenticates Client

Authentication

Authentication Conversions

Optional PAC refrash

EAP Success

Protected Data Session

WPA or CCKM Key Management

Data

EAP-FAST Advantages **RFC 4851**

-Supports Windows single sign-on and password expiration
-Wide range of OS support
-Does not require certificates or PKI
-Full support for 802.11i, 802.11x, TKIP and AES
-Support for WDS and CCKM
-Resistant to dictionary attacks

Weaknesses

-PAC can be intercepted and used to compromise credentials
-Rouge AP with same SSID could be used to inject a new PAC which could be used to obtain username and a cleartext password (EAP-FAST w/GTC) or launch a dictionary attack

Source: Layer3.wordpress.com

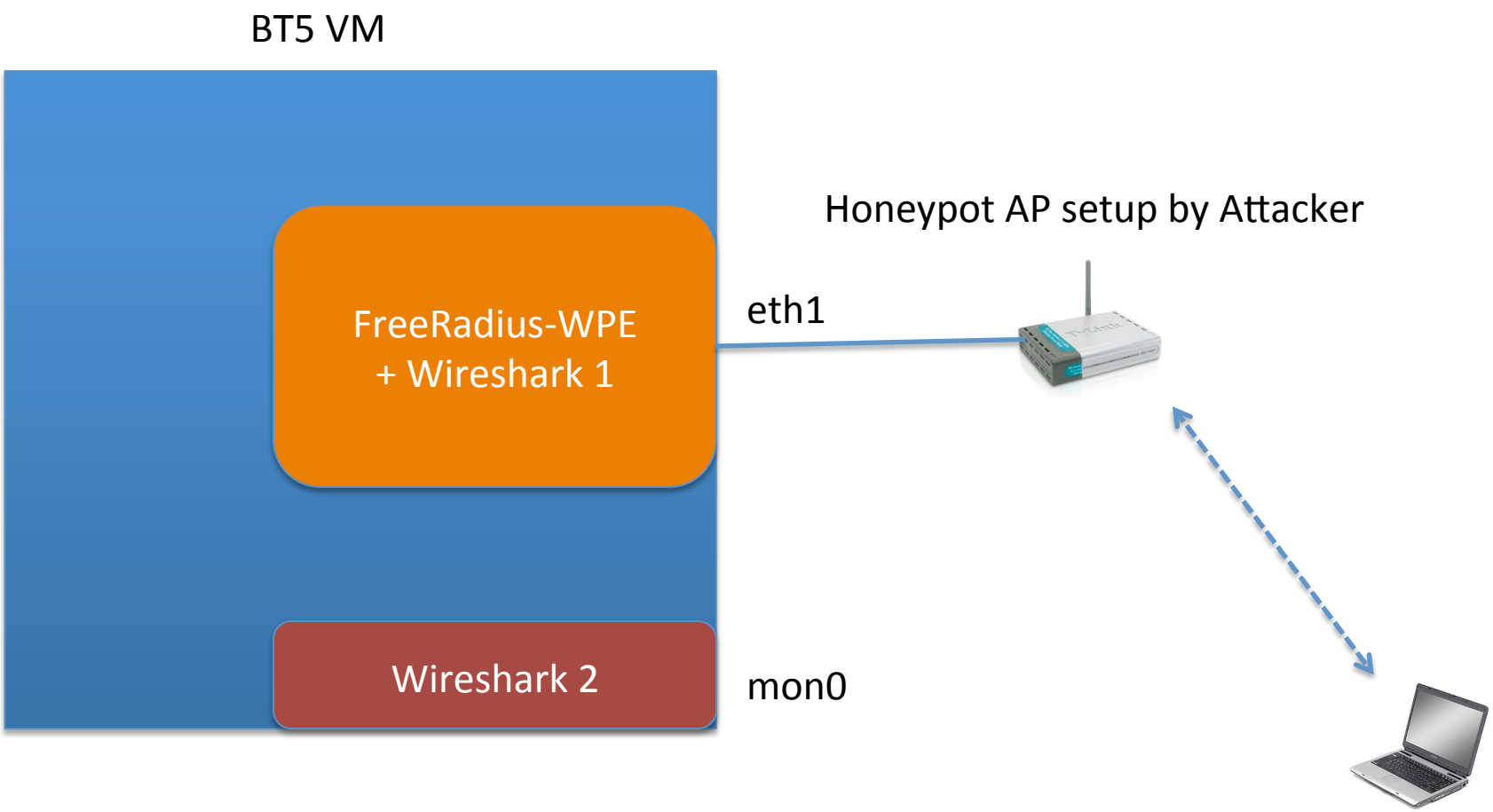
PEAP Reloaded

- PEAP is the most popular enterprise Wi-Fi security mechanism used
- We will setup PEAP using FreeRadius-WPE and do a full authentication test
- We will use OS X in this video
- Next we will explore the security risks using both Windows and OS X

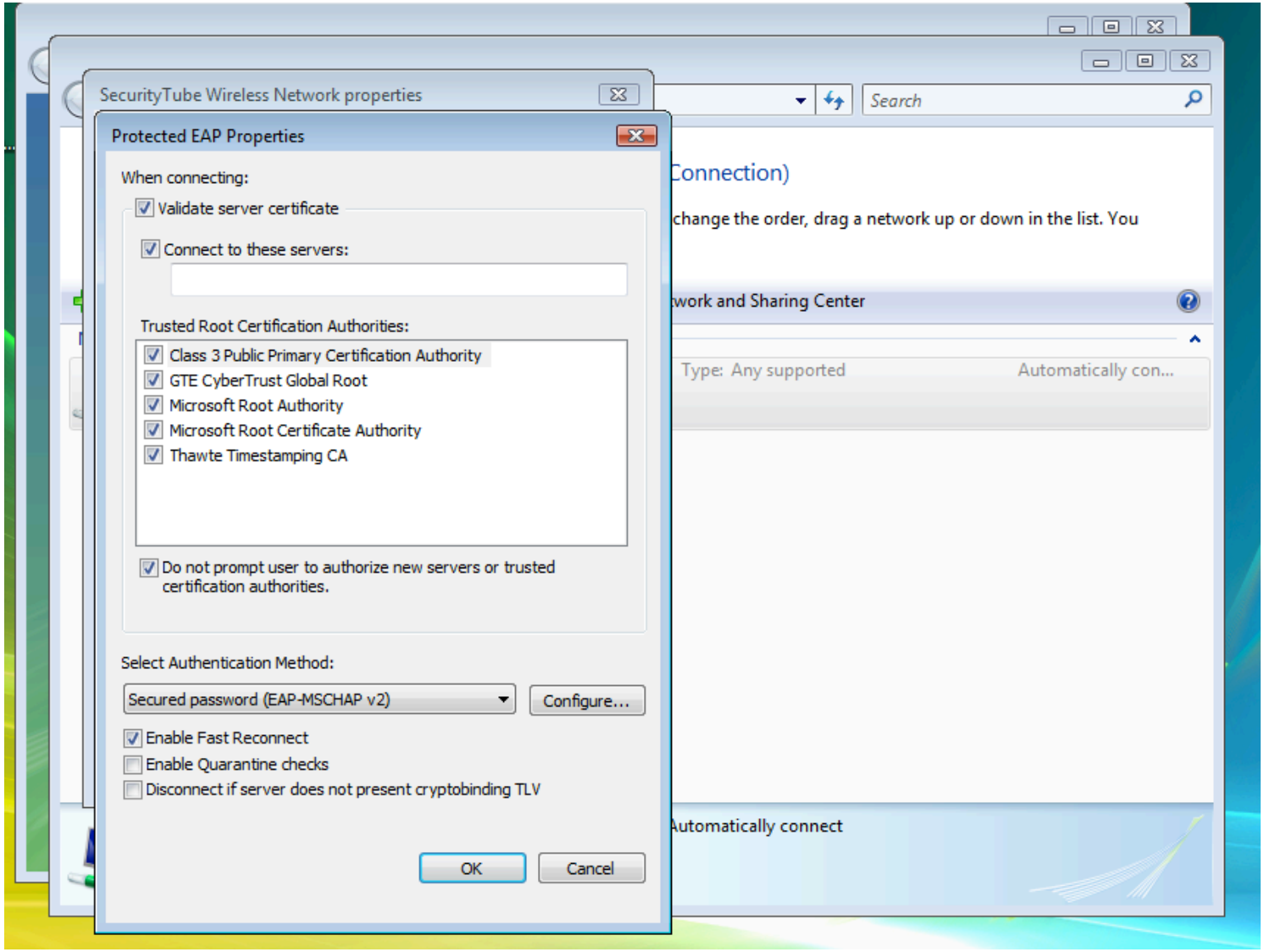
Understanding the Insecurity

- Server side certificates
  - Fake ones can be created
  - Clients may not prompt or user may accept invalid certificates
- Setup a Honeypot with FreeRadius-WPE
  - Client connects
  - Accepts fake certificate
  - Sends authentication details over MSCHAPv2 in the TLS tunnel
  - Attacker's radius server logs these details
  - Apply dictionary / reduced possibility bruteforce attack using Asleap by Joshua Wright

Network Architecture



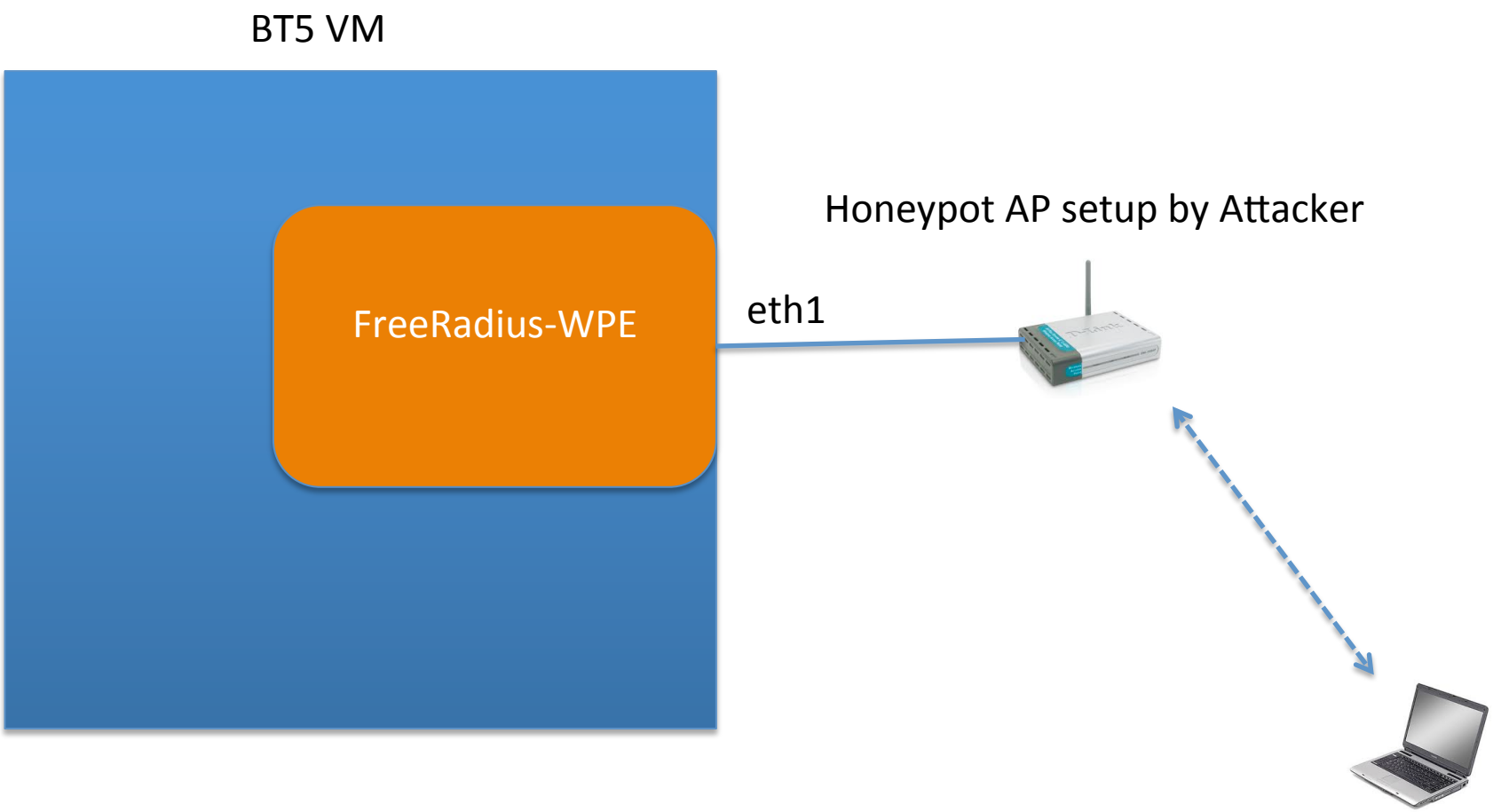
Windows PEAP Hacking Summed Up in 1 Slide ☺



Inner Authentication in EAP-TTLS

- MSCHAPv2
- MSCHAP
- CHAP
- PAP
- …

Network Architecture



Insecure Configuration in 3rd Party Utils

- Most client cards ship with drivers + utility
- Utility typically created by card manufacturer and sometimes provides more information
  - Signal strength
  - Operation mode choice
  - …
- Configuring network security with these utilities is a bad idea
  - Can lead to insecure configurations
- Solution: Use your OS's inbuilt utility and configure it securely

Please turn in your completed feedback form at the registration desk.

Q&A

Questions