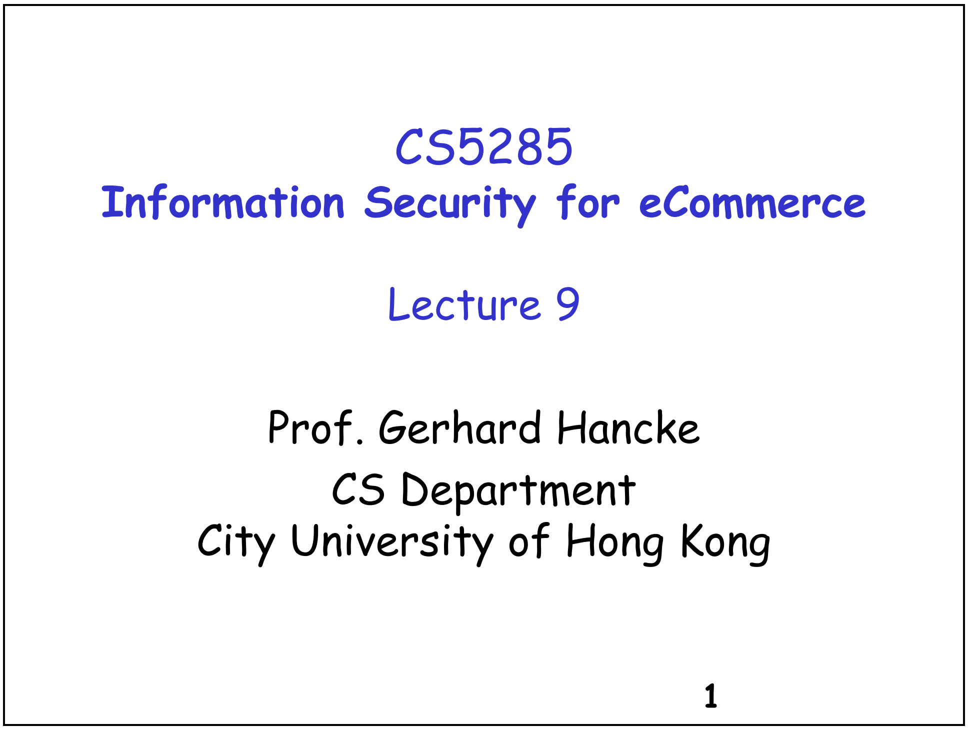
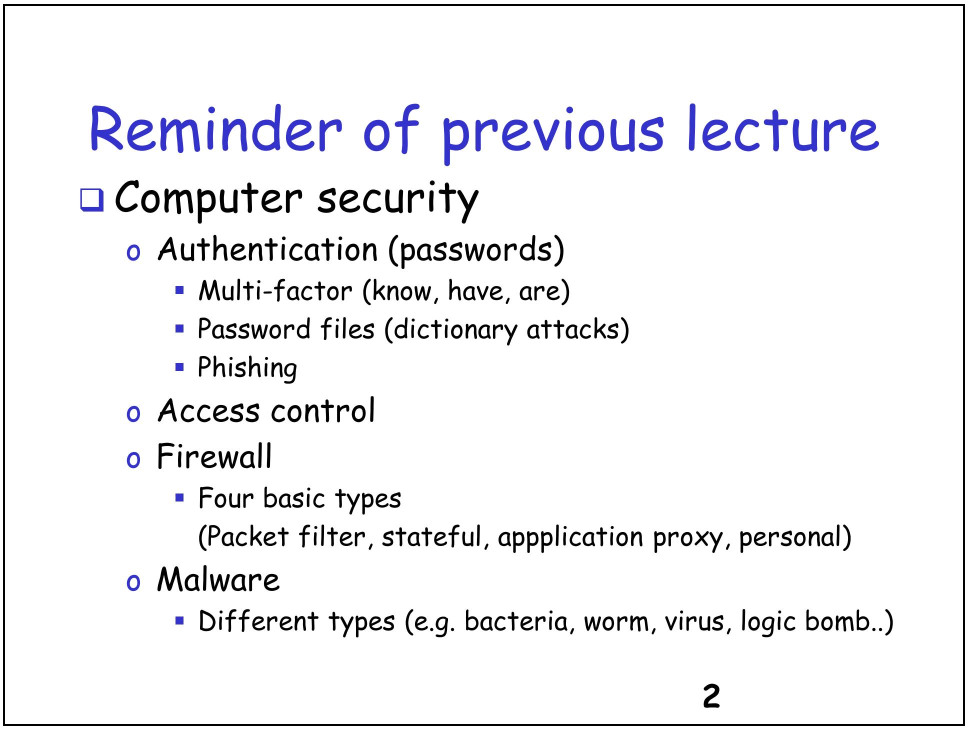


CS5285
Information Security for eCommerce

Lecture 9

Prof. Gerhard Hancke
CS Department
City University of Hong Kong



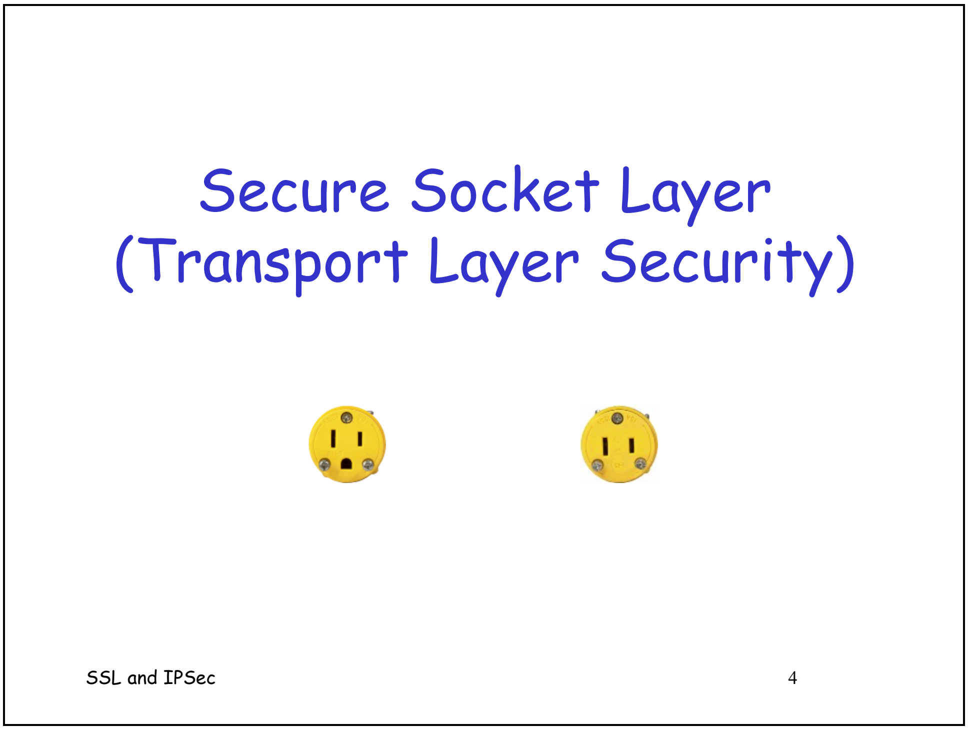
Reminder of previous lecture

- Computer security
  - Authentication (passwords)
    - Multi-factor (know, have, are)
    - Password files (dictionary attacks)
    - Phishing
  - Access control
  - Firewall
    - Four basic types
(Packet filter, stateful, appplication proxy, personal)
  - Malware
    - Different types (e.g. bacteria, worm, virus, logic bomb..)

Today's Lecture

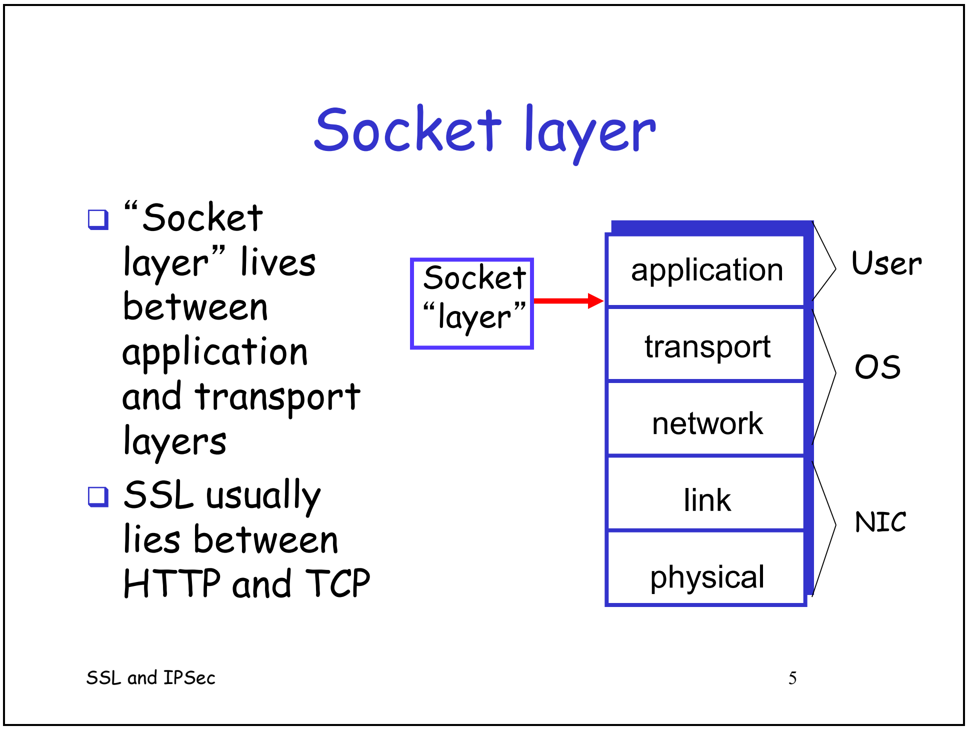
- For all e-commerce systems we need to securely communicate and exchange data
- Aspects of Network Security
  - Web (TLS/SSL)/IPSEC
  - WiFi/Mobile Networks
  - DoS
- CILO1,CILO2, CILO3 and CILO4

  (Data security, security requirements, security measures, security assessment)



Secure Socket Layer (Transport Layer Security)

You must know the idea of SSL well.



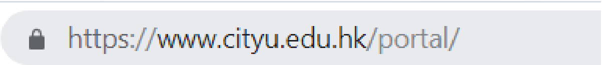
Socket layer

- "Socket layer" lives between application and transport layers
- SSL usually lies between HTTP and TCP

You should know where SSL works as opposed to where IPSec works.

What is SSL?

- ❑ Secure Socket Layer (SSL) is the protocol used for most secure transactions over the Internet

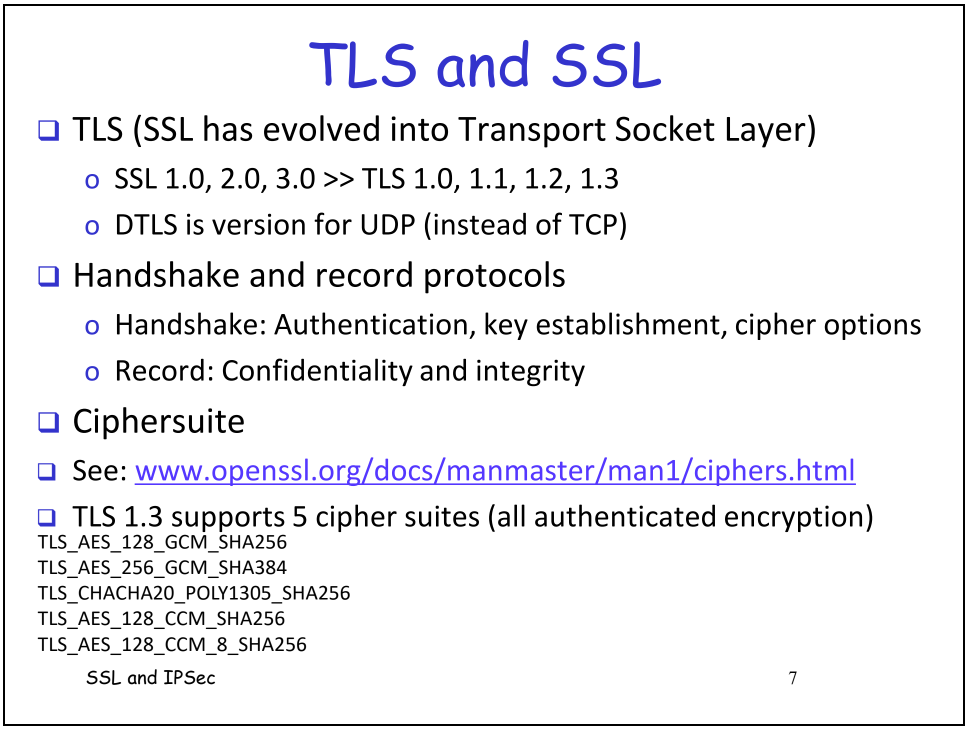


https://www.cityu.edu.hk/portal/

- ❑ For example, if you want to buy a book at amazon.com…
  - o You want to be sure that you are dealing with Amazon (**one-way authentication**)
  - o Your credit card information must be protected in transit (**data confidentiality**)
  - o As long as you have money, Amazon doesn't care who you are (**authentication need not to be mutual**)
    - ▪ Mutual version does exist (if client has certificate, server to server)

You should know what security services SSL provides by default. You must also be able to identify which parts of the protocol provide these services.

What do we need when we do Internet security?



TLS and SSL

- ❑ TLS (SSL has evolved into Transport Socket Layer)
  - o SSL 1.0, 2.0, 3.0 >> TLS 1.0, 1.1, 1.2, 1.3
  - o DTLS is version for UDP (instead of TCP)
- ❑ Handshake and record protocols
  - o Handshake: Authentication, key establishment, cipher options
  - o Record: Confidentiality and integrity
- ❑ Ciphersuite
- ❑ See: www.openssl.org/docs/manmaster/man1/ciphers.html
- ❑ TLS 1.3 supports 5 cipher suites (all authenticated encryption)

TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384
TLS_CHACHA20_POLY1305_SHA256
TLS_AES_128_CCM_SHA256
TLS_AES_128_CCM_8_SHA256

SSL and IPSec 7

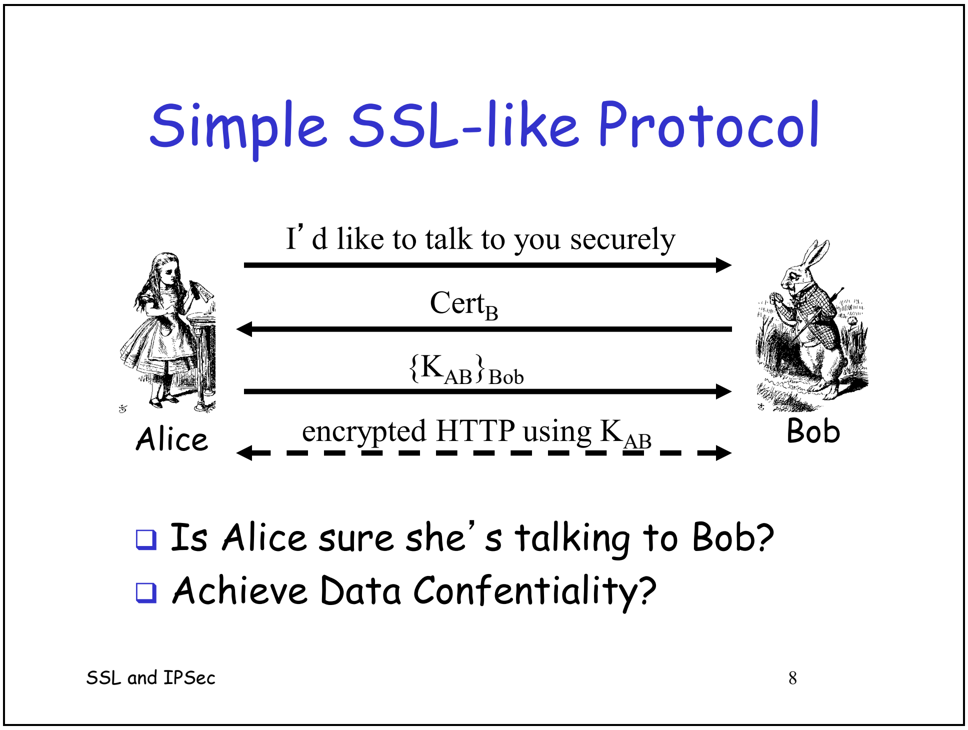
For interest

Proposed for TLS 1.3 to have no mode with encryption that do not also have message authentication

For example, TLS 1.2 has TLS_RSA_WITH_AES_128_CBC_SHA256 (that hash the data but has no MAC).

Galois Counter Mode (GCM)

Encrypt in counter, CBC-MAC for message authentication (CCM)



Simple SSL-like Protocol

I’ d like to talk to you securely

Cert_B

$\{K_{AB}\}_{Bob}$

encrypted HTTP using K_{AB}

Alice

Bob

- ❑ Is Alice sure she’ s talking to Bob?
- ❑ Achieve Data Confentiality?

This is the basic idea behind SSL.

Alice establishes a shared key with Bob, and the fact that Bob can decrypt and use the key means we have a fresh value (the key) and something only Bob can do (decrypt the key with his private key) so we are sure we are currently talking to Bob.

Who is authenticated?

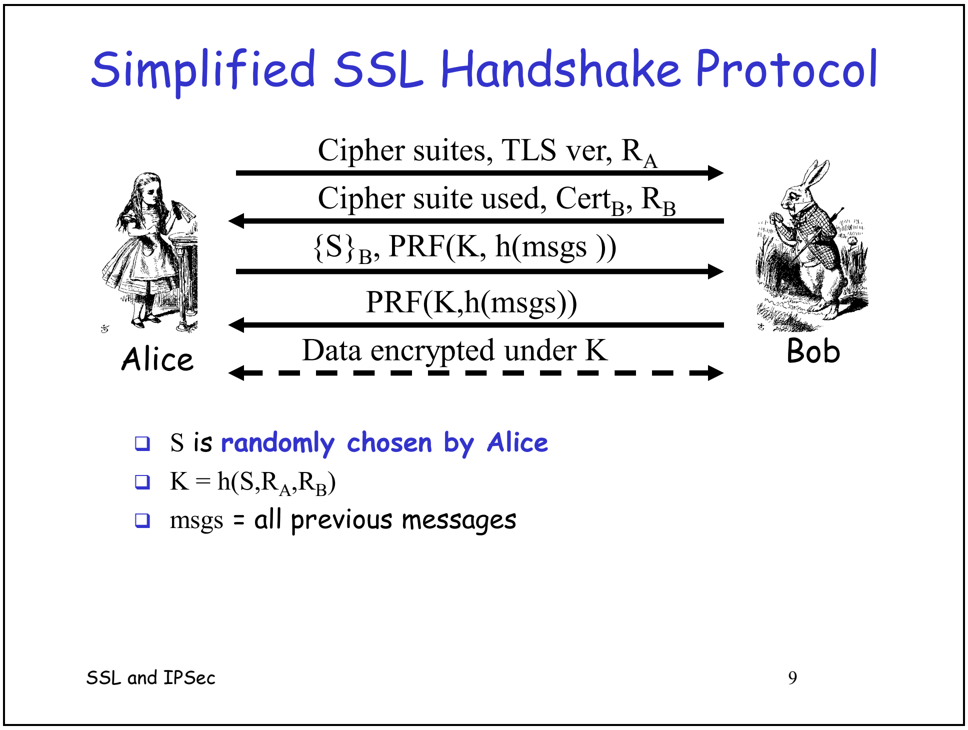
What is the purpose?

Do we have our three security properties?

Bob authenticated

Alice not

Data confidentiality



Simplified SSL Handshake Protocol

Alice

Bob

Cipher suites, TLS ver, R_A

Cipher suite used, $Cert_B$, R_B

$\{S\}_B$, PRF(K, h(msgs))

PRF(K,h(msgs))

Data encrypted under K

- S is **randomly chosen by Alice**
- $K = h(S,R_A,R_B)$
- msgs = all previous messages

SSL and IPSec 9

'msgs' is the previous message exchanged between Alice and Bob in the handshake.

Authentication: Bob to Alice

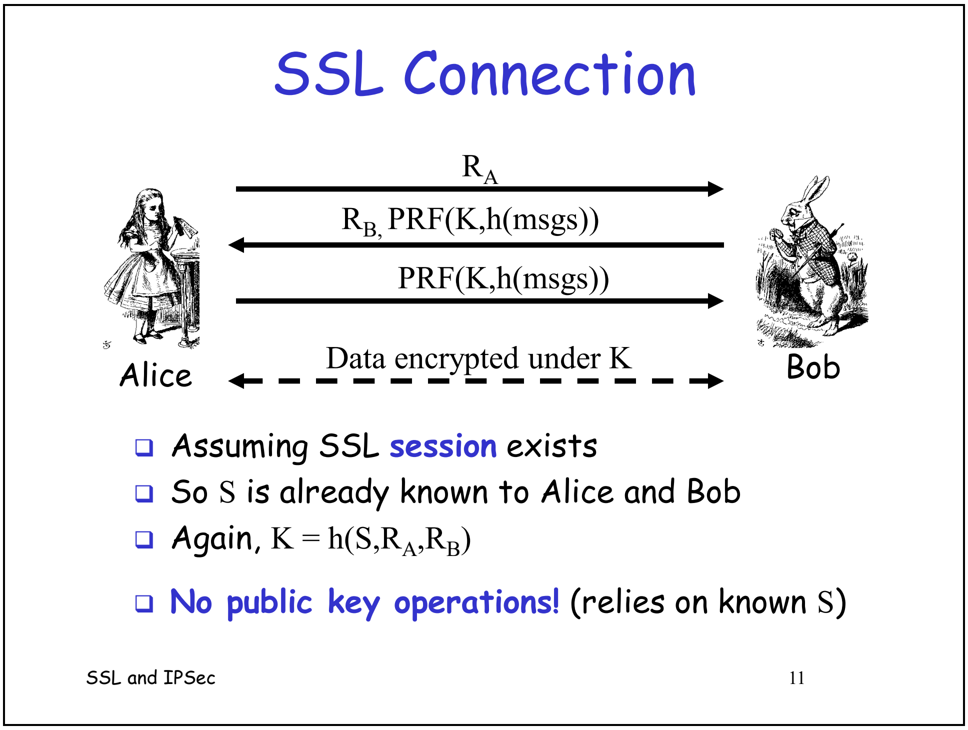
Key control: Both

Key confirmation: Yes

PRF is pseudo-random functions (like a hash). These are the "Finished" messages confirming key and authentication complete. Each side must send and verify other side Finished message before data transfer starts.

SSL Sessions vs Connections

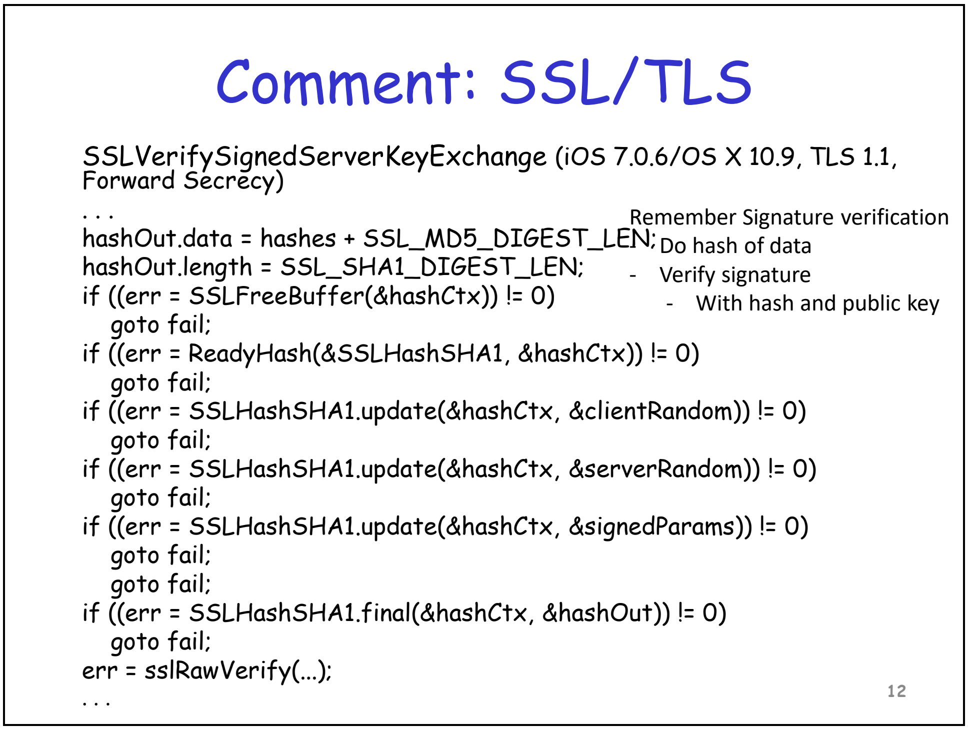
- SSL designed for use with HTTP 1.0
- HTTP 1.0 usually opens multiple simultaneous (parallel) **connections**
- SSL session establishment is costly
  - Due to public key operations
- SSL has an efficient protocol for opening new connections given an existing session



SSL Connection

- Assuming SSL **session** exists
- So S is already known to Alice and Bob
- Again, $K = h(S,R_A,R_B)$
- **No public key operations!** (relies on known S)

S is the same as for the initial handshake.



Comment: SSL/TLS

SSLVerifySignedServerKeyExchange (iOS 7.0.6/OS X 10.9, TLS 1.1, Forward Secrecy)

```
. . .
hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
hashOut.length = SSL_SHA1_DIGEST_LEN;
if ((err = SSLFreeBuffer(&hashCtx)) != 0)
    goto fail;
if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;
err = sslRawVerify(...);
. . .
```

Remember Signature verification

Do hash of data

- Verify signature
  - With hash and public key

For interest

This function will return err (fail if anything but equal to 0).

The second goto will always execute (and err =0 because of the condition before it being valid). Therefore sslRawVerify never has an effect.

https://nakedsecurity.sophos.com/2014/02/24/anatomy-of-a-goto-fail-apples-ssl-bug-explained-plus-an-unofficial-patch/

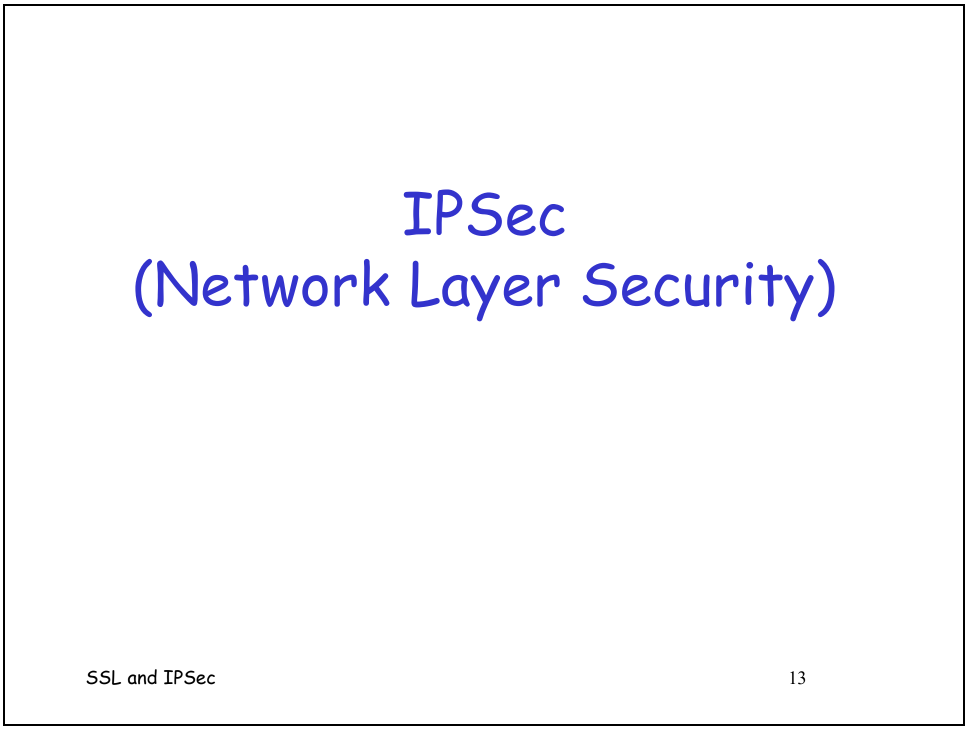
Attacker can use this if:

Trick you into visting an imposter HTTPS site, e.g. by using a poisoned public Wi-Fi access point.

Force your browser (or other software) into using specific cipher option (forward secrecy), possible because the server decides what encryption algorithms it will support.

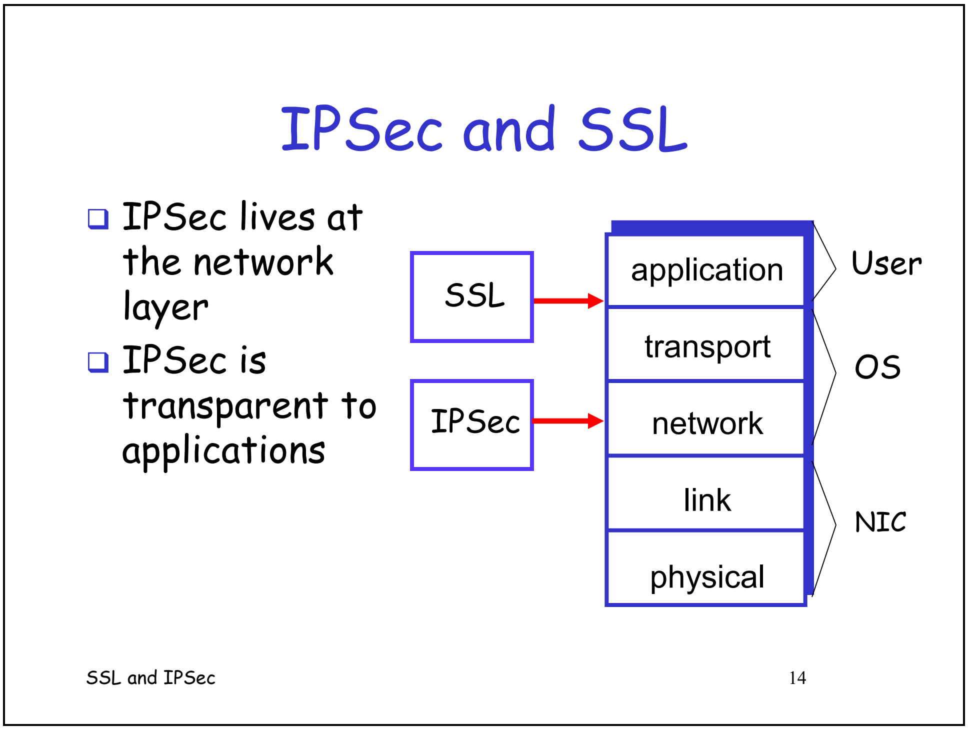
Force your browser (or other software) into using TLS 1.1, possible because the server decides what TLS versions it will allow.

Supply a legitimate-looking TLS certificate with a mismatched private key.



IPSec
(Network Layer Security)

IPSec as a whole is also important.



IPSec and SSL

- IPSec lives at the network layer
- IPSec is transparent to applications

SSL

IPSec

application

transport

network

link

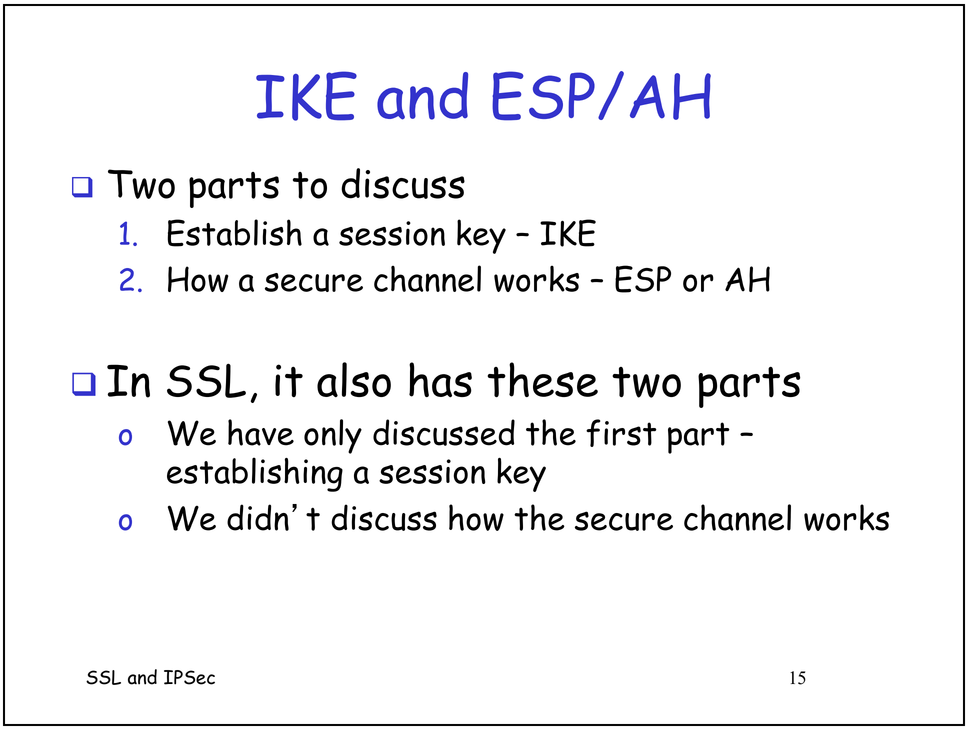
physical

User

OS

NIC

Once again you should think where SSL works on the network protocol stack in relation to IPSec



IKE and ESP/AH

- Two parts to discuss
  1. Establish a session key – IKE
  2. How a secure channel works – ESP or AH
- In SSL, it also has these two parts
  - We have only discussed the first part – establishing a session key
  - We didn't discuss how the secure channel works

IPSec has lots of different aspects – do not get confused!!!

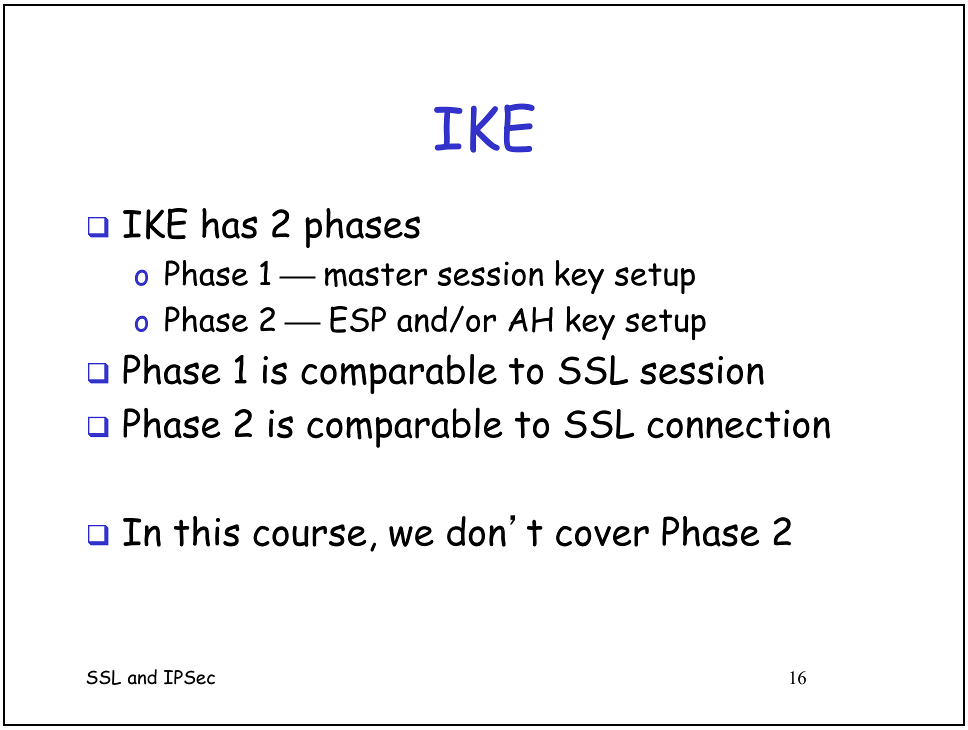
Separate IKE from ESP/AH.

IKE Internet Key Exchange is only to enable the two parties to set up a shared key!

This should not be completely new to you as we already had a lecture on key management! This is simply application of what you studied already.

ESP – Encapsulating Security Payload

AH- Authentication Header

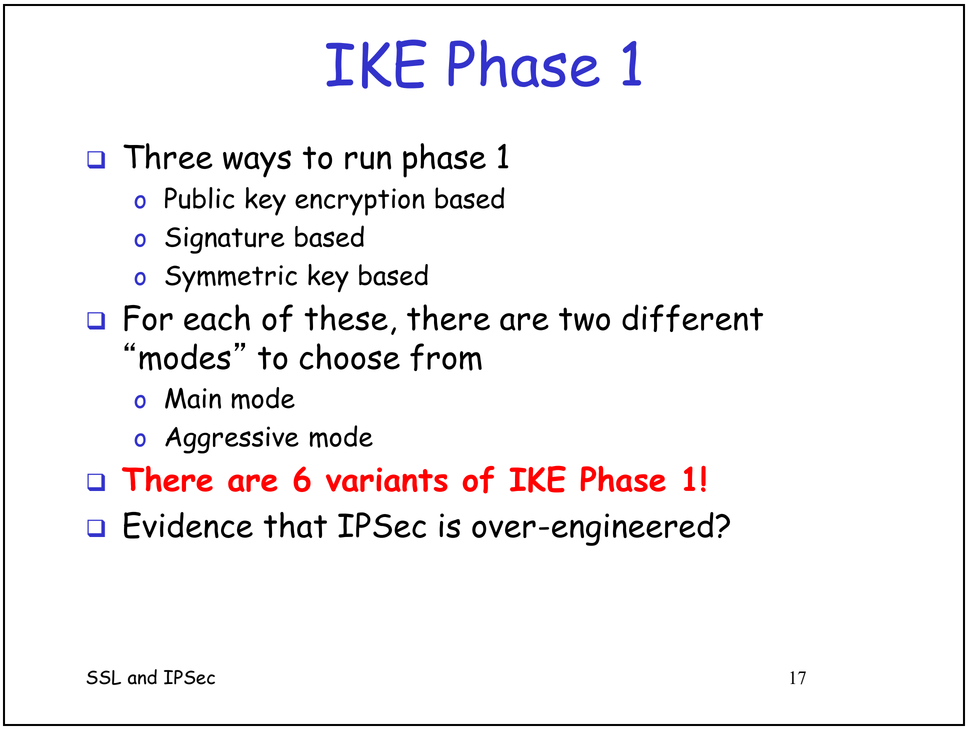


IKE

- ❑ IKE has 2 phases
  - o Phase 1 — master session key setup
  - o Phase 2 — ESP and/or AH key setup
- ❑ Phase 1 is comparable to SSL session
- ❑ Phase 2 is comparable to SSL connection

- ❑ In this course, we don't cover Phase 2

For interest.



IKE Phase 1

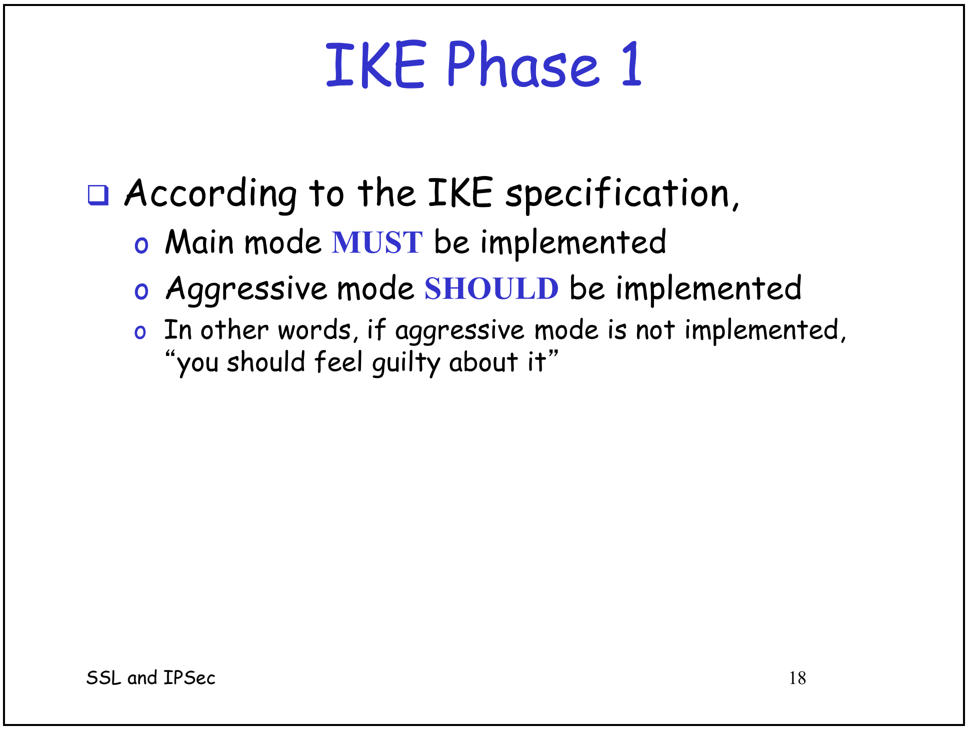
- Three ways to run phase 1
  - Public key encryption based
  - Signature based
  - Symmetric key based
- For each of these, there are two different "modes" to choose from
  - Main mode
  - Aggressive mode
- **There are 6 variants of IKE Phase 1!**
- Evidence that IPSec is over-engineered?

We are only concerned with IKE Phase 1

You must know the 6 variants – you do not need to memorise them. I will not ask you: What does Aggressive mode of IKE using symmetric key look like?

You could be given the protocol and asked to explain how it works.

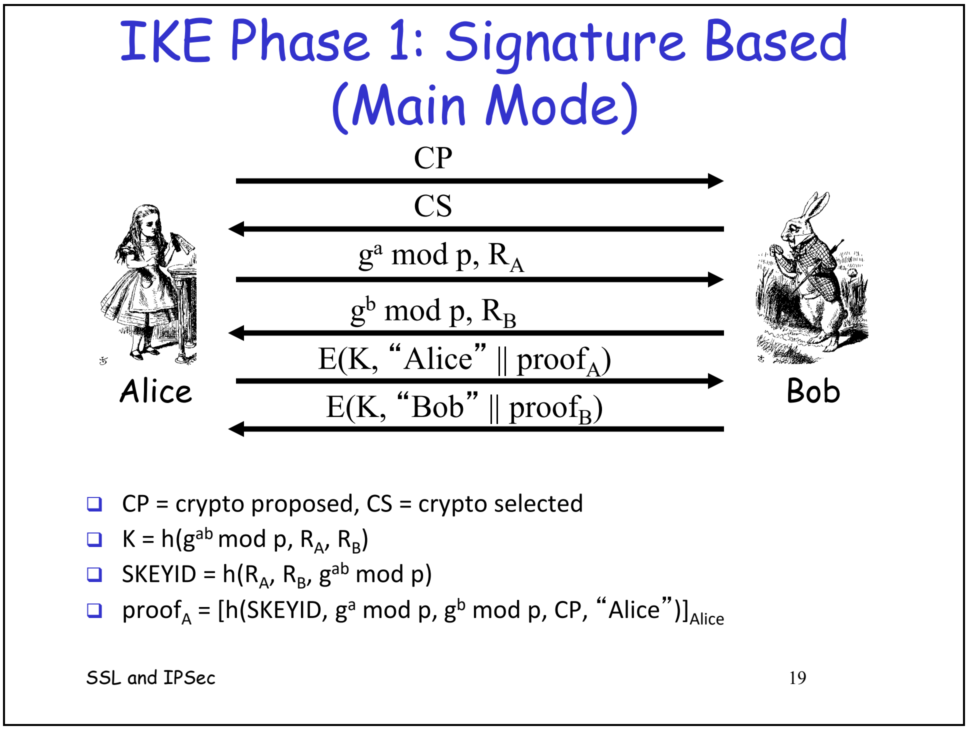
I might ask you what the main difference between main and aggressive mode is.



IKE Phase 1

- According to the IKE specification,
  - Main mode **MUST** be implemented
  - Aggressive mode **SHOULD** be implemented
  - In other words, if aggressive mode is not implemented, "you should feel guilty about it"

For interest only.



IKE Phase 1: Signature Based (Main Mode)

CP

CS

$g^a \bmod p, R_A$

$g^b \bmod p, R_B$

$E(K, \text{“Alice”} \| \text{proof}_A)$

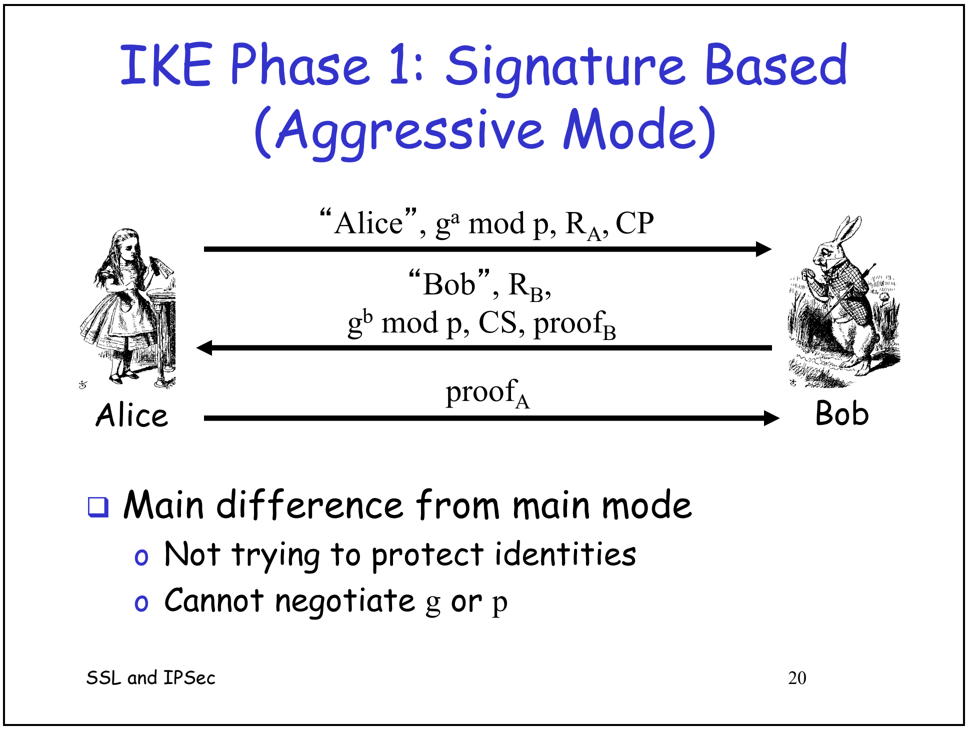
$E(K, \text{“Bob”} \| \text{proof}_B)$

- CP = crypto proposed, CS = crypto selected
- $K = h(g^{ab} \bmod p, R_A, R_B)$
- $\text{SKEYID} = h(R_A, R_B, g^{ab} \bmod p)$
- $\text{proof}_A = [h(\text{SKEYID}, g^a \bmod p, g^b \bmod p, \text{CP}, \text{“Alice”})]_{\text{Alice}}$

Keep the following in mind: Each of the protocols use the basic Diffie-Hellman approach!! The reference to signature, public key encryption or symmetric key encryption is relevant to how the Proof is calculated!

Remember that with DH we had a problem with man in the middle (remember Trudy?). So the proof is there to make sure the DH exchange did actually take place between Alice and Bob.

Why can generate K and SKEYID? (only Alice and Bob – what about MITM? T and A , and T and B). Who can generate proof A or B? Only A or B. T cannot generate the proof.



IKE Phase 1: Signature Based (Aggressive Mode)

“Alice”, g^a mod p, R_A, CP

“Bob”, R_B, g^b mod p, CS, proof_B

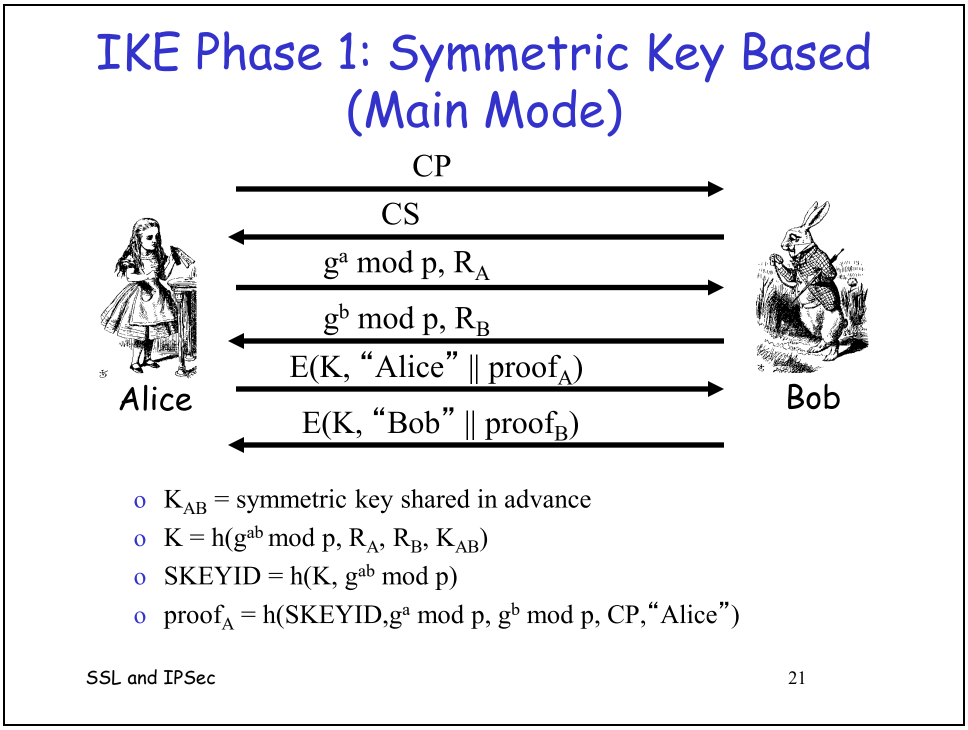
proof_A

- Main difference from main mode
  - Not trying to protect identities
  - Cannot negotiate g or p

Note that Alice and Bob ID now no longer private.

Also not explicit key authentication.

These differences are in general



IKE Phase 1: Symmetric Key Based (Main Mode)

Alice → Bob: CP

Bob → Alice: CS

Alice → Bob: g^a mod p, R_A

Bob → Alice: g^b mod p, R_B

Alice → Bob: E(K, "Alice" || proof_A)

Bob → Alice: E(K, "Bob" || proof_B)

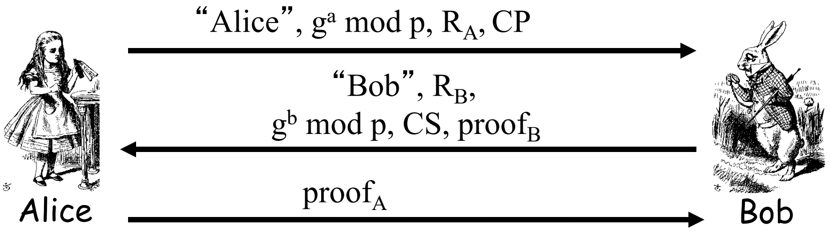
- K_{AB} = symmetric key shared in advance
- K = h(g^{ab} mod p, R_A, R_B, K_{AB})
- SKEYID = h(K, g^{ab} mod p)
- proof_A = h(SKEYID,g^a mod p, g^b mod p, CP,"Alice")

How does Bob know he is talking to Alice?

Problems with Symmetric Key Based (Main Mode)

- ❑ Catch
  - o Alice sends her ID in message 5
  - o Alice's ID encrypted with K
  - o To find K Bob must know K_{AB}
  - o To get K_{AB} Bob must know he's talking to Alice!
- ❑ Result: **Alice's ID must be IP address!**
- ❑ Useless mode for the "road warrior"

IKE Phase 1: Symmetric Key Based (Aggressive Mode)

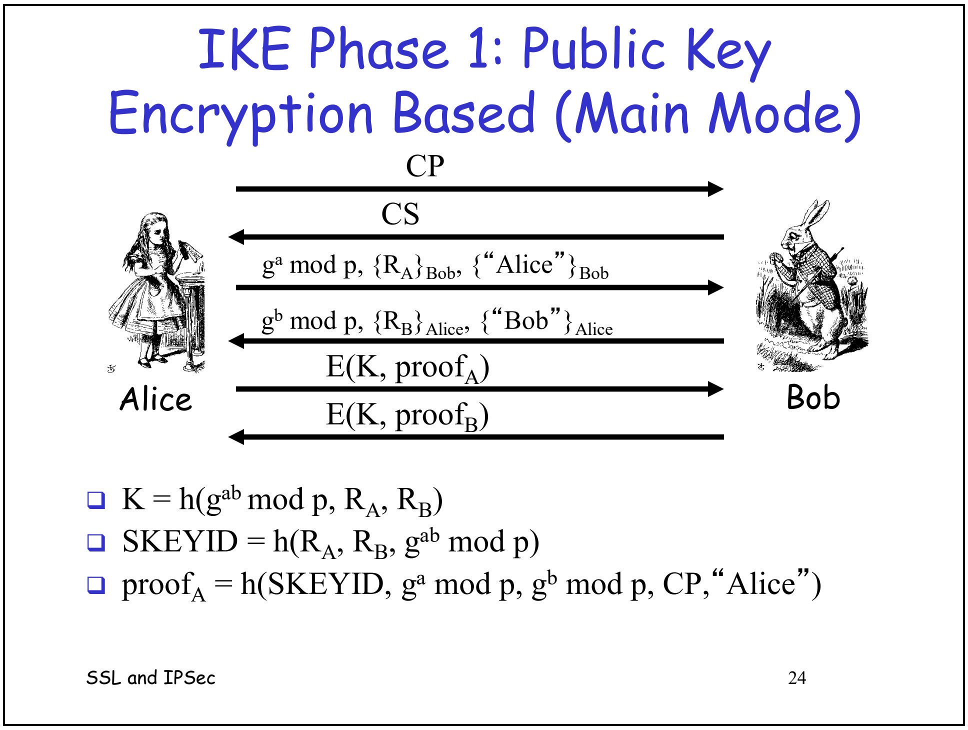


"Alice", g^a mod p, R_A, CP

"Bob", R_B,
g^b mod p, CS, proof_B

proof_A

- Same format as digital signature aggressive mode
- Not trying to hide identities…
- As a result, does **not** have problems of main mode



IKE Phase 1: Public Key Encryption Based (Main Mode)

CP

CS

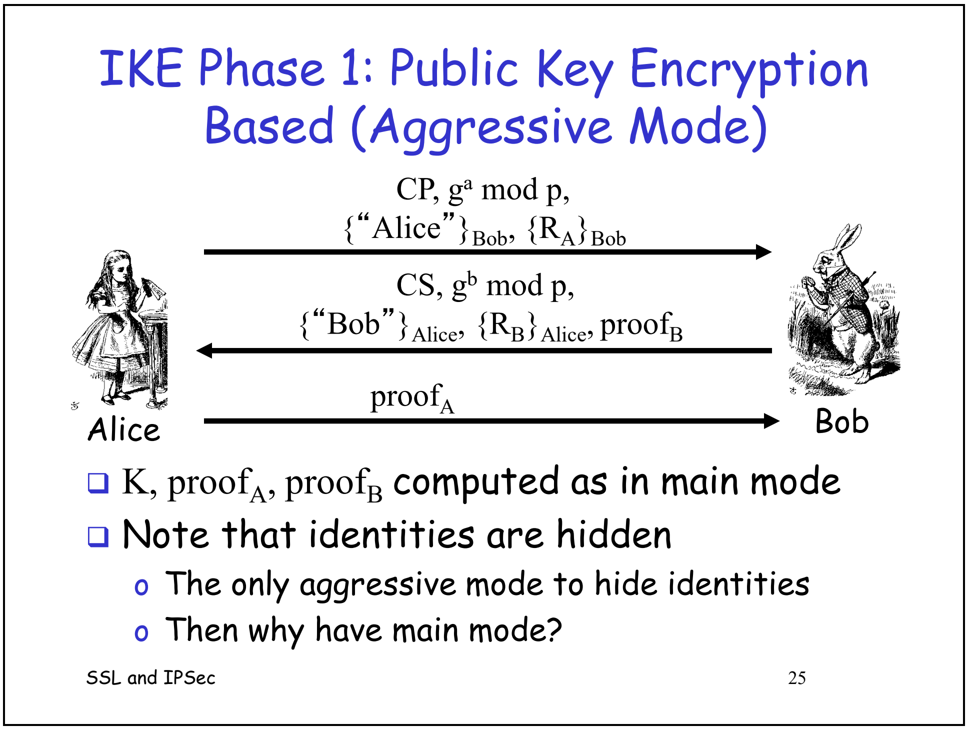
g^a mod p, $\{R_A\}_{Bob}$, $\{$“Alice”$\}_{Bob}$

g^b mod p, $\{R_B\}_{Alice}$, $\{$“Bob”$\}_{Alice}$

E(K, $proof_A$)

E(K, $proof_B$)

- K = h(g^{ab} mod p, R_A, R_B)
- SKEYID = h(R_A, R_B, g^{ab} mod p)
- $proof_A$ = h(SKEYID, g^a mod p, g^b mod p, CP,“Alice”)



IKE Phase 1: Public Key Encryption Based (Aggressive Mode)

CP, g^a mod p,
{"Alice"}$_{Bob}$, {R_A}$_{Bob}$

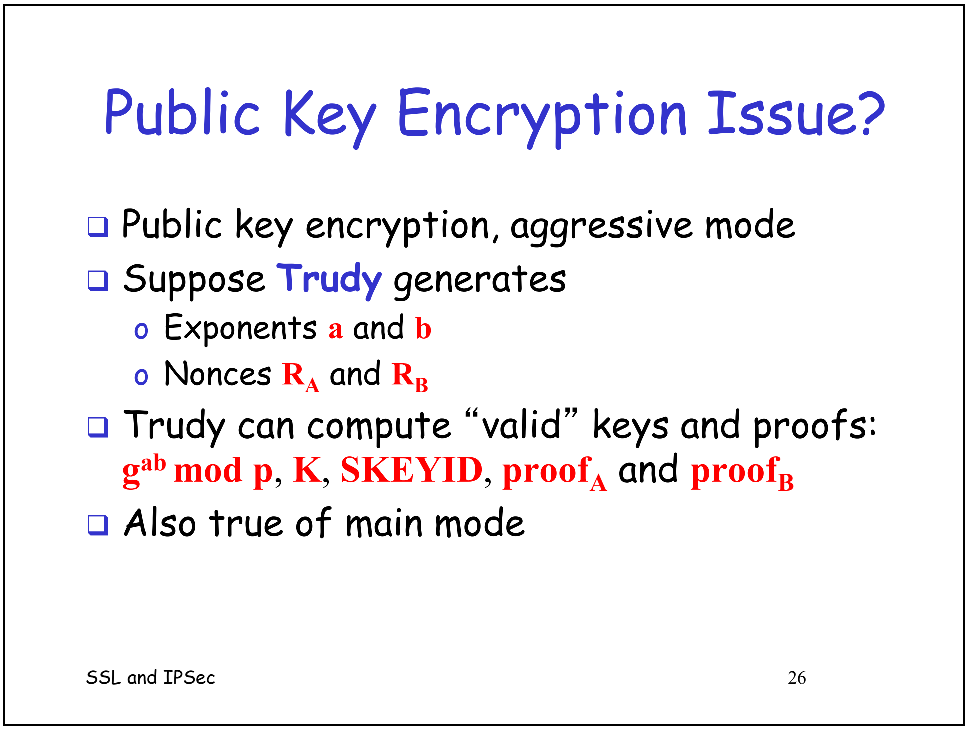
CS, g^b mod p,
{"Bob"}$_{Alice}$, {R_B}$_{Alice}$, $proof_B$

$proof_A$

Alice

Bob

- K, $proof_A$, $proof_B$ computed as in main mode
- Note that identities are hidden
  - The only aggressive mode to hide identities
  - Then why have main mode?

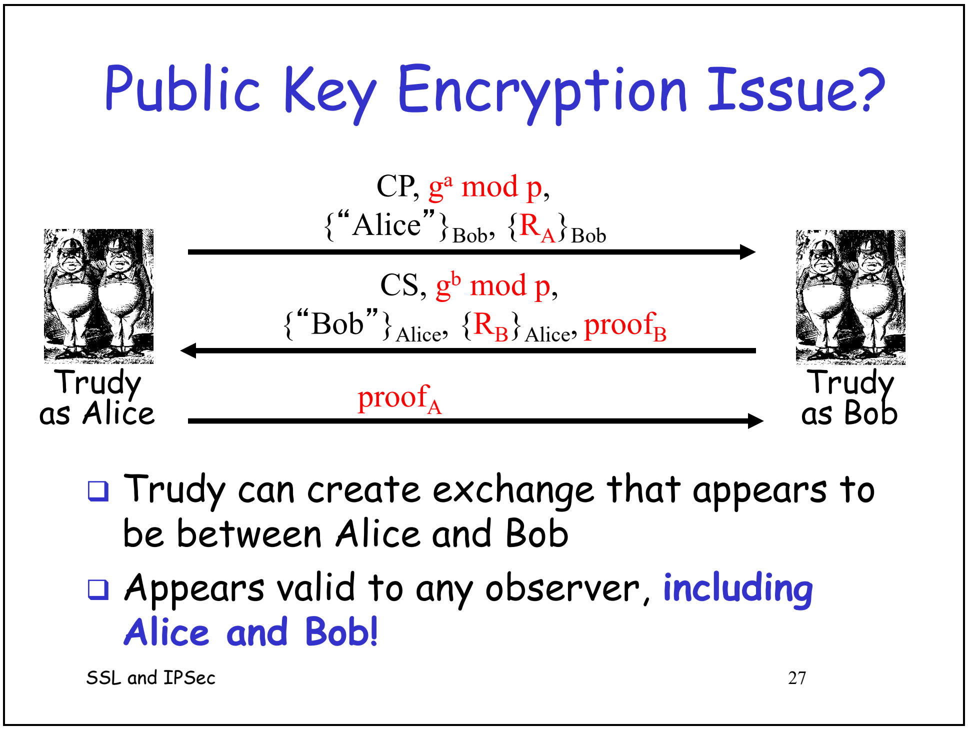


Public Key Encryption Issue?

- Public key encryption, aggressive mode
- Suppose **Trudy** generates
  - Exponents **a** and **b**
  - Nonces $\mathbf{R_A}$ and $\mathbf{R_B}$
- Trudy can compute “valid” keys and proofs: $\mathbf{g^{ab}}$ **mod p**, **K**, **SKEYID**, $\mathbf{proof_A}$ and $\mathbf{proof_B}$
- Also true of main mode

Sldie 26-28 for interest only

This is not man in the middle, it just means you can pretend that A and B had a conversation.



Public Key Encryption Issue?

CP, $g^a \bmod p$, $\{\text{“Alice”}\}_{\text{Bob}}$, $\{R_A\}_{\text{Bob}}$

CS, $g^b \bmod p$, $\{\text{“Bob”}\}_{\text{Alice}}$, $\{R_B\}_{\text{Alice}}$, proof_B

proof_A

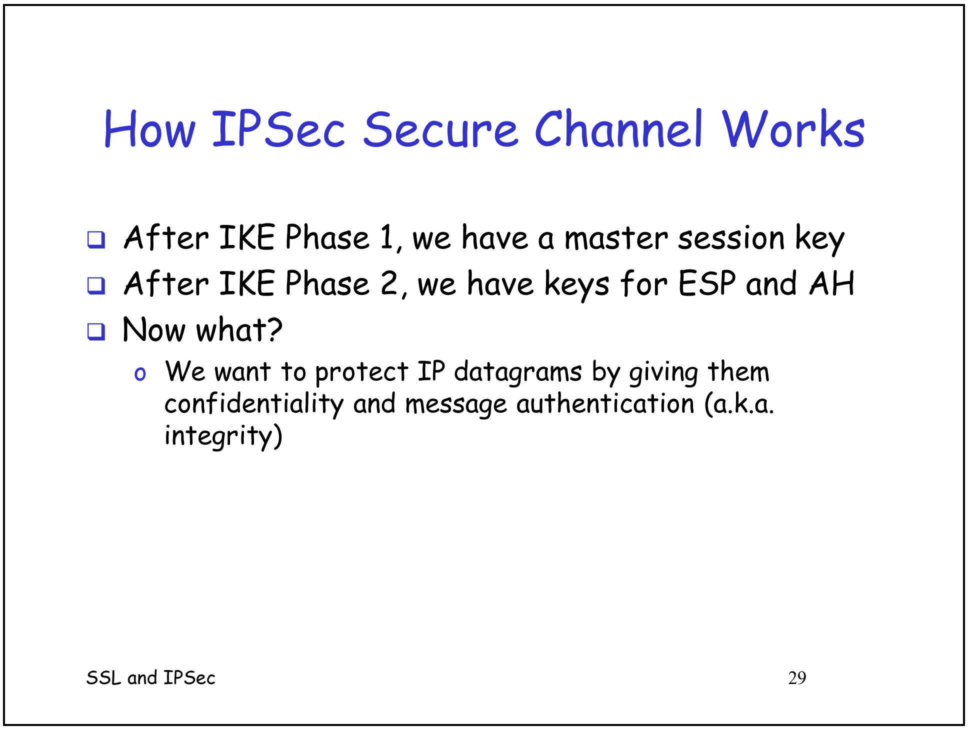
- Trudy can create exchange that appears to be between Alice and Bob
- Appears valid to any observer, **including Alice and Bob!**

SSL and IPSec 27

Not participant…the protocol exchange looks valid (even if verifier by Alice or Bob later)

Plausible Deniability

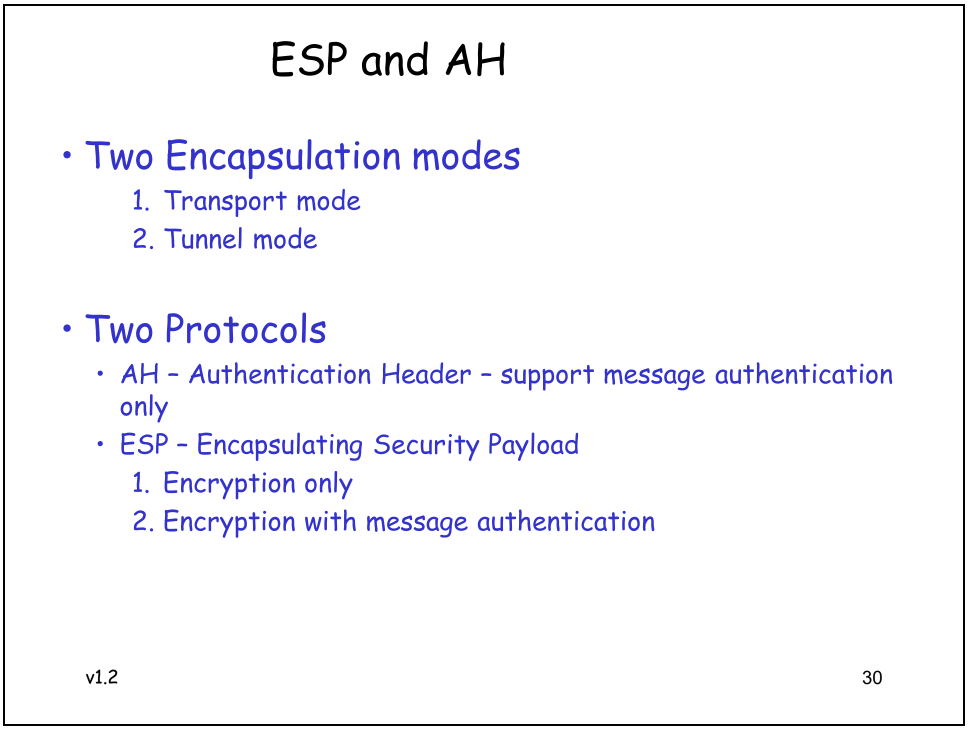
- A security failure?
- In this mode of IPSec, it is a feature!
  - **Plausible deniability:** Alice and Bob can deny that any conversation has taken place!
- In some cases it might be a security failure
  - If Alice makes a purchase from Bob, she could later repudiate it (unless she had signed)



How IPSec Secure Channel Works

- After IKE Phase 1, we have a master session key
- After IKE Phase 2, we have keys for ESP and AH
- Now what?
  - We want to protect IP datagrams by giving them confidentiality and message authentication (a.k.a. integrity)

For interest only.



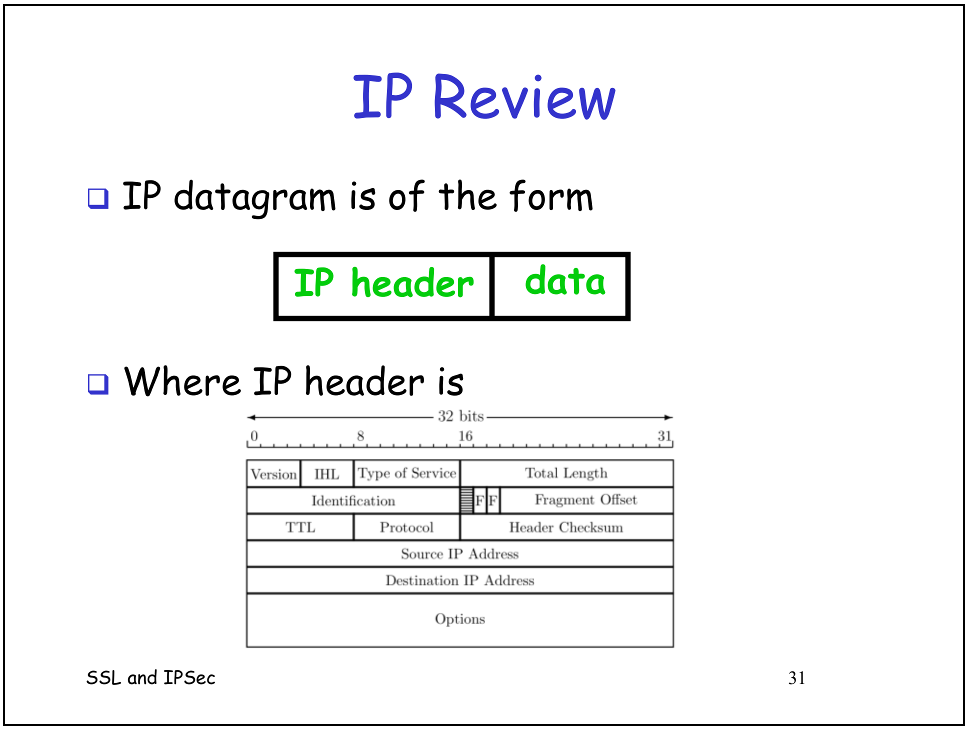
ESP and AH

- Two Encapsulation modes
  1. Transport mode
  2. Tunnel mode

- Two Protocols
  - AH – Authentication Header – support message authentication only
  - ESP – Encapsulating Security Payload
    1. Encryption only
    2. Encryption with message authentication

See this as the second part of IPSec.

You must be able to distinguish between the two encapsulation modes and the two protocols AH and ESP



IP Review

- IP datagram is of the form

IP header	data

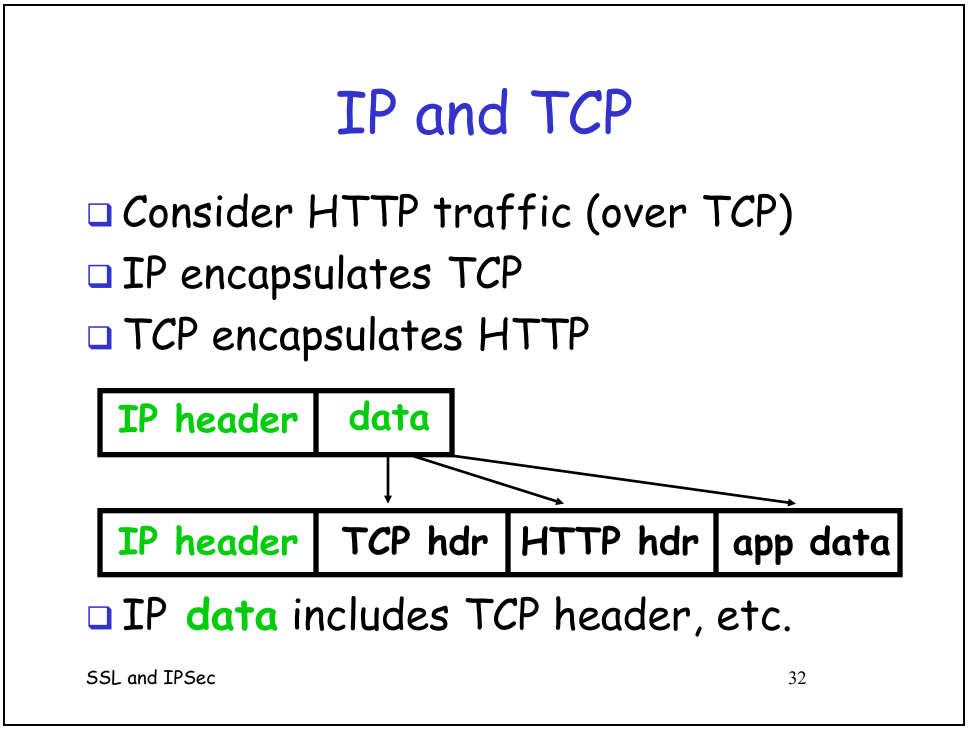
- Where IP header is

32 bits

0 8 16 31

Version	IHL	Type of Service	Total Length			
Identification		▤	F	F	Fragment Offset	
TTL	Protocol	Header Checksum				
Source IP Address						
Destination IP Address						
Options						

No need to remember low level details of IP header. You are interested most in the fact that there is an IP header and where it is, whether it is the original or a new IPSec header and the source/destination IP address.



IP and TCP

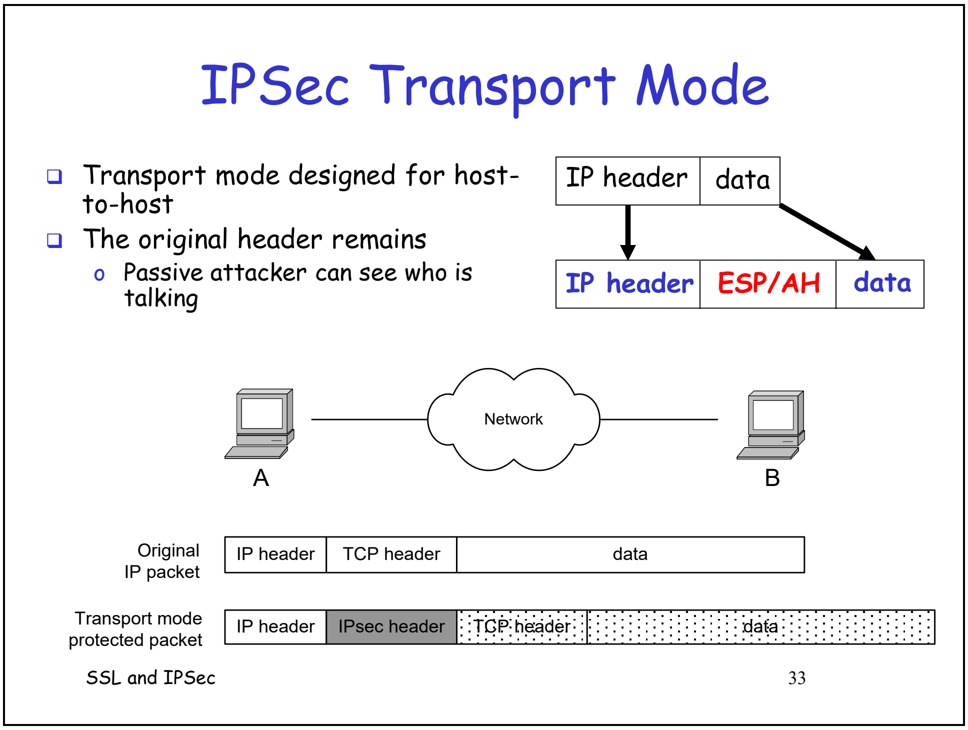
- Consider HTTP traffic (over TCP)
- IP encapsulates TCP
- TCP encapsulates HTTP

IP header	data

IP header	TCP hdr	HTTP hdr	app data

- IP **data** includes TCP header, etc.

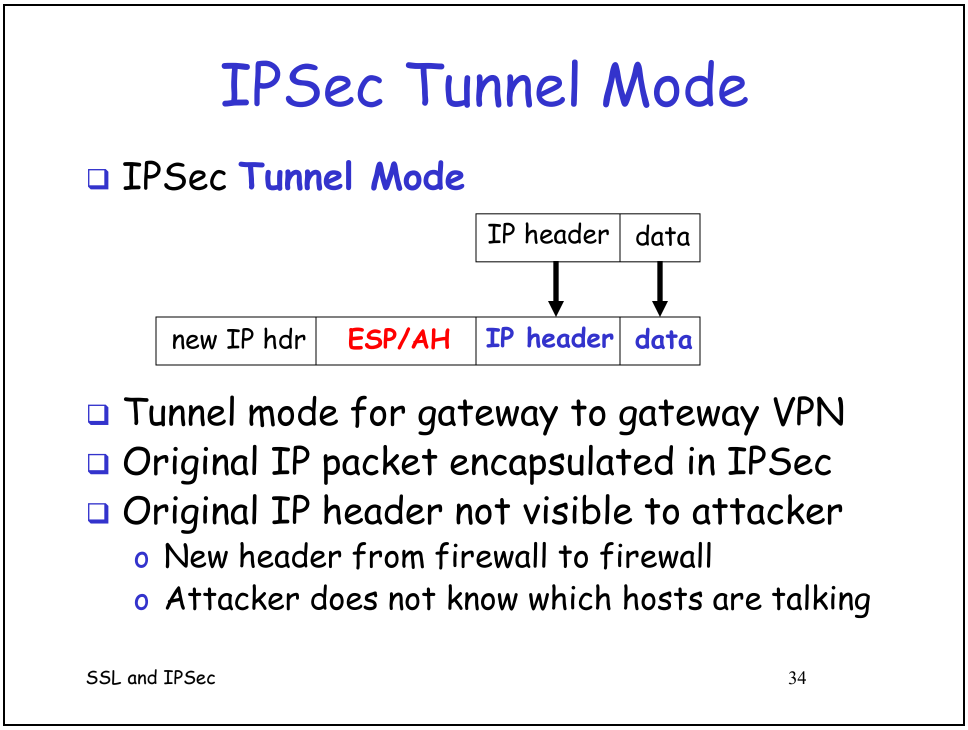
Background. From IPSec perspective everything above network layer is just data.



IPSec Transport Mode

- Transport mode designed for host-to-host
- The original header remains
  - Passive attacker can see who is talking

In transport mode we send the original IP header as is. This means that even in ESP mode the original IP header is not encrypted and we can see who is the actual sender receiver is.



IPSec Tunnel Mode

- IPSec **Tunnel Mode**

IP header | data

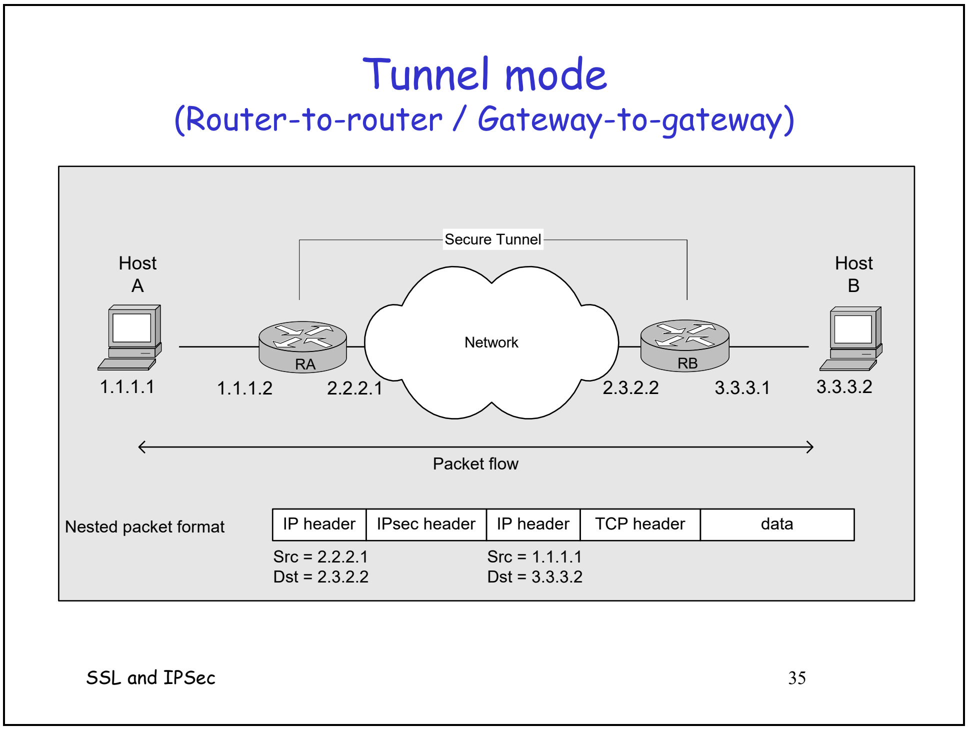
new IP hdr | ESP/AH | IP header | data

- Tunnel mode for gateway to gateway VPN
- Original IP packet encapsulated in IPSec
- Original IP header not visible to attacker
  - New header from firewall to firewall
  - Attacker does not know which hosts are talking

In tunnel mode IPSec runs between two IPSec devices. The original IP header is seen as data and then a new one is added.

So if Alice wants to send a packet to Bob the IP header (source: Alice, target: Bob) is data and is replaced with a new one – source IPSec router 1, target: IPSec router 2.

We cannot just encrypt IP headers as we wish – remember this packet still needs to go across a public network where devices need to see where it is going. Not all routeres belong to us and they cannot all decrypt/encrypt headers as neede.d



Tunnel mode

(Router-to-router / Gateway-to-gateway)

Nested packet format

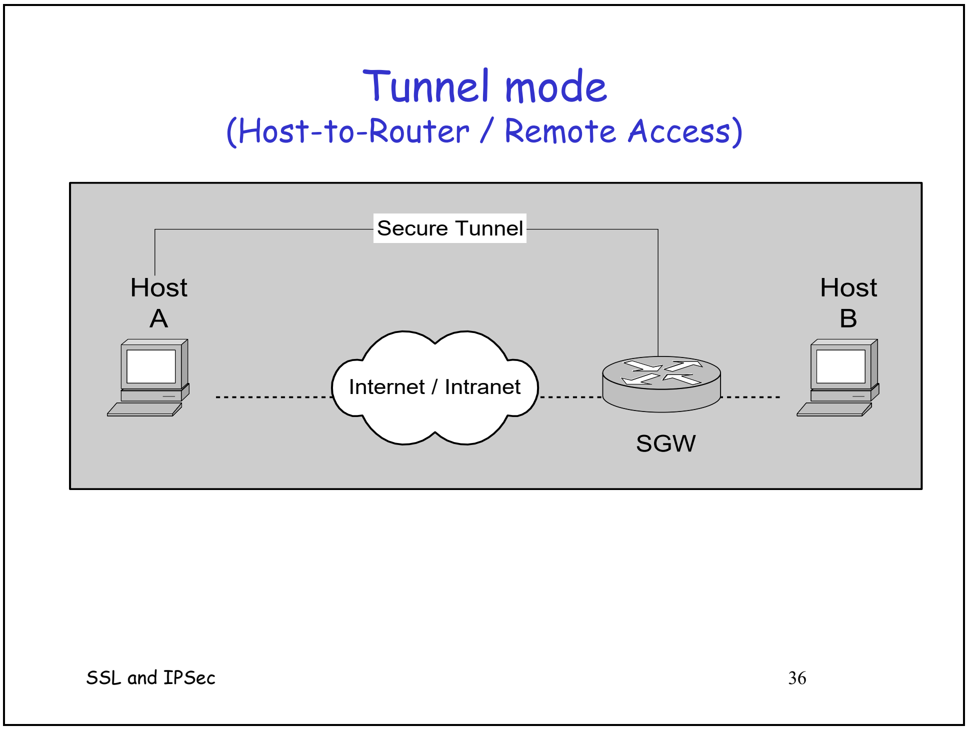
IP header	IPsec header	IP header	TCP header	data

Src = 2.2.2.1
Dst = 2.3.2.2
		Src = 1.1.1.1
		Dst = 3.3.3.2		

- Make sure you understand this example of tunnel mode.

- What is your source and destination in each case? What does IPSec look like?

- Description of the network (two companies communicating over the Internet
- Secure tunnel between GWs
- Host A generates a packet
- The RA GW encapsulate the packet adding a new IP header
- The RB GW de-capsulate the packet (strip extra headers) and injects the packet to B's network
- Note that the routing tables for A and B needs to direct the packets to the GWs so these could be directed



Tunnel mode

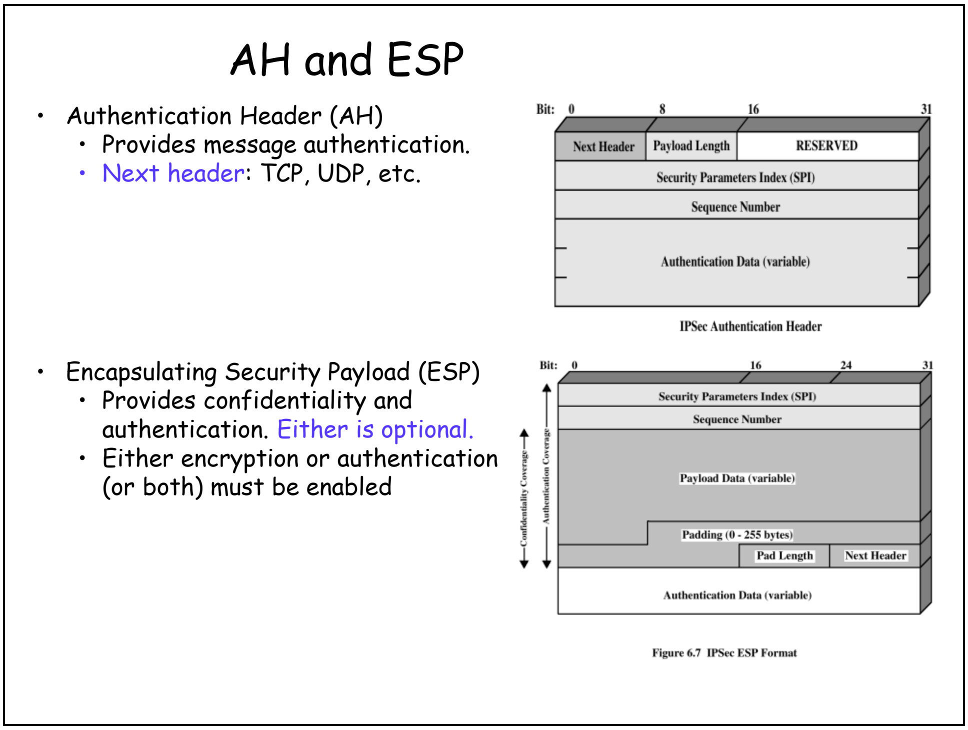
(Host-to-Router / Remote Access)

•Example: a worker working from home…

•Note that Host A imitates the role of a SGW like in the former slide
it behaves as a SGW and as a Host

•Note that host A has now two interfaces so it can fully imitate the GW
(it has one physical and one virtual)

•All the packet flow/structure should look the same



AH and ESP

- Authentication Header (AH)
  - Provides message authentication.
  - Next header: TCP, UDP, etc.

Bit:
0
8
16
31
Next Header
Payload Length
RESERVED
Security Parameters Index (SPI)
Sequence Number
Authentication Data (variable)

IPSec Authentication Header

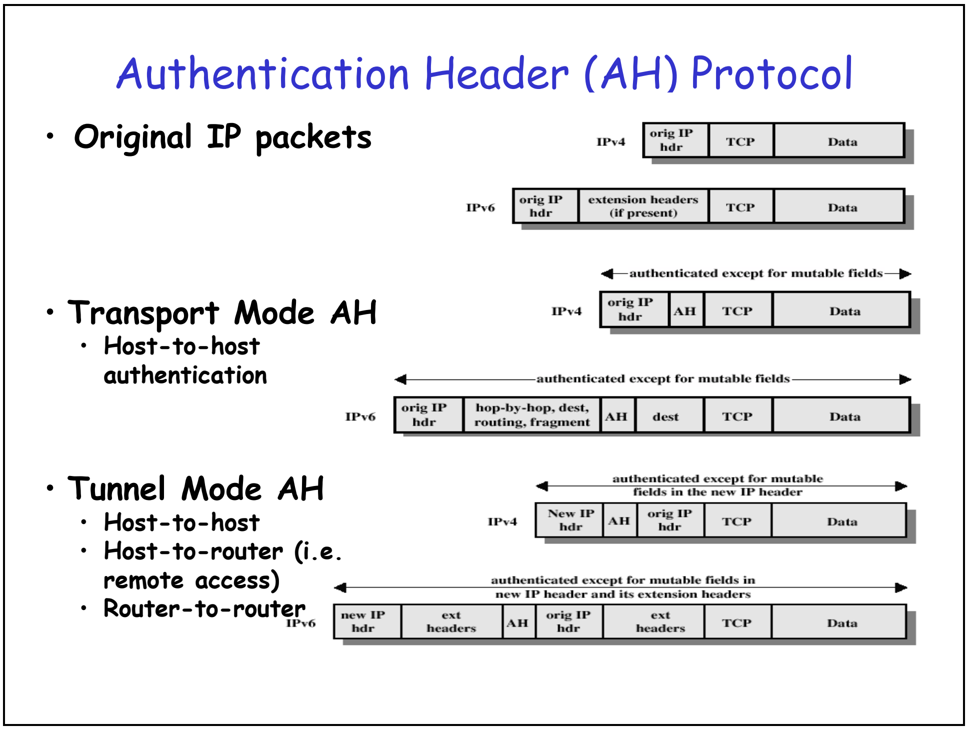
- Encapsulating Security Payload (ESP)
  - Provides confidentiality and authentication. Either is optional.
  - Either encryption or authentication (or both) must be enabled

Bit:
0
16
24
31
Security Parameters Index (SPI)
Sequence Number
Payload Data (variable)
Padding (0 - 255 bytes)
Pad Length
Next Header
Authentication Data (variable)
Confidentiality Coverage
Authentication Coverage

Figure 6.7 IPSec ESP Format

You must know what services AH and ESP provide.

Packets formats on this slide for interest.



Authentication Header (AH) Protocol

- Original IP packets

IPv4: orig IP hdr | TCP | Data

IPv6: orig IP hdr | extension headers (if present) | TCP | Data

- Transport Mode AH
  - Host-to-host authentication

authenticated except for mutable fields

IPv4: orig IP hdr | AH | TCP | Data

authenticated except for mutable fields

IPv6: orig IP hdr | hop-by-hop, dest, routing, fragment | AH | dest | TCP | Data

- Tunnel Mode AH
  - Host-to-host
  - Host-to-router (i.e. remote access)
  - Router-to-router

authenticated except for mutable fields in the new IP header

IPv4: New IP hdr | AH | orig IP hdr | TCP | Data

authenticated except for mutable fields in new IP header and its extension headers

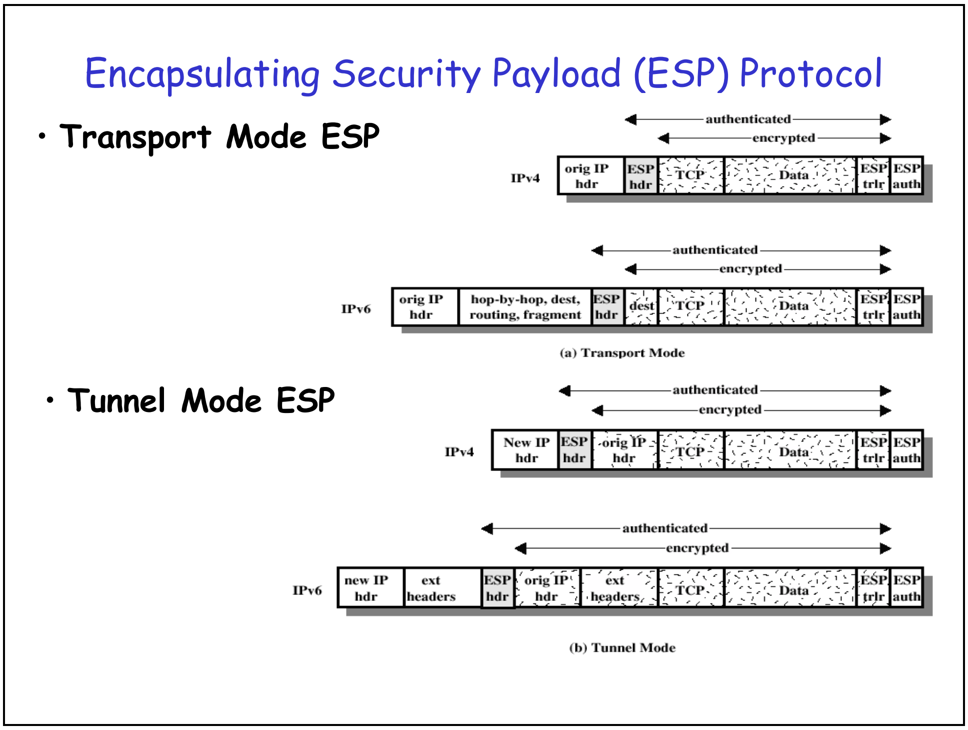
IPv6: new IP hdr | ext headers | AH | orig IP hdr | ext headers | TCP | Data

You need to study slide 38 and 39 in detail.

At first AH tunnel mode looks a bit useless – the original IP header is replaced but then you keep it plaintext (AH only does data origin authentication).

The benefits of AH in tunnel mode is that the entire original IP header is integrity protected – not only the immutable fields. In the new IP header the mutable fields (variable that change during transmission are not protected).

Note that in AH protocol authenticated extends to the entire packet including the IP header used to route the message.



Encapsulating Security Payload (ESP) Protocol

- **Transport Mode ESP**

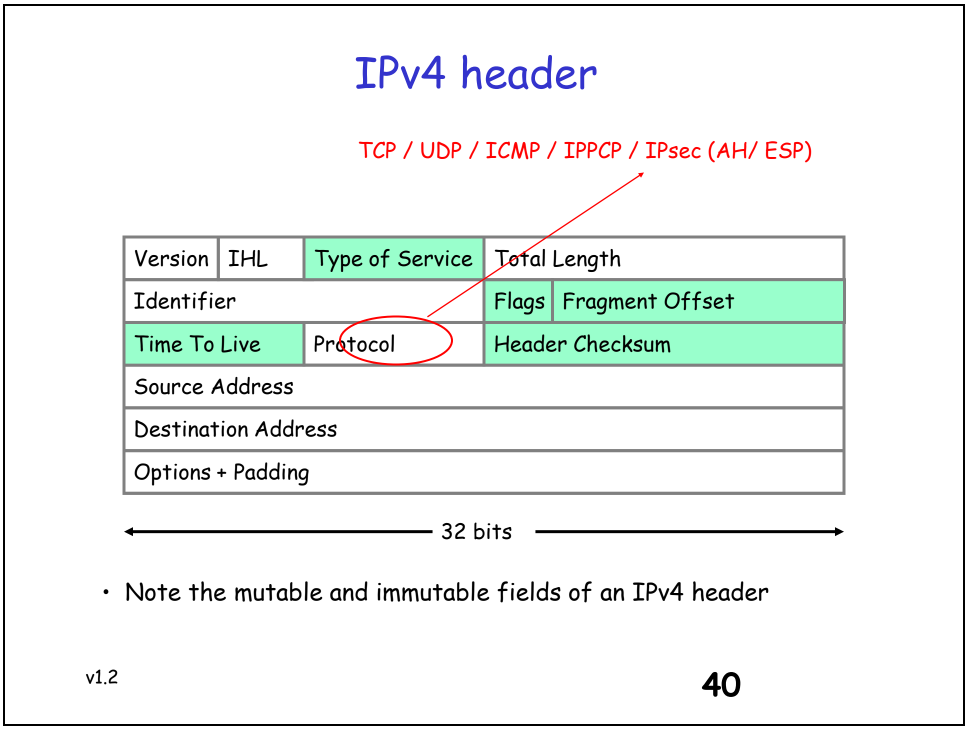
(a) Transport Mode

- **Tunnel Mode ESP**

(b) Tunnel Mode

Note that the IP header is not protected in any way here.

Hoever, the original IP header in tunnel mode is now confidential.



IPv4 header

TCP / UDP / ICMP / IPPCP / IPsec (AH/ ESP)

<table>
<tr><td>Version</td><td>IHL</td><td colspan="2">Type of Service</td><td colspan="4">Total Length</td></tr>
<tr><td colspan="4">Identifier</td><td>Flags</td><td colspan="3">Fragment Offset</td></tr>
<tr><td colspan="2">Time To Live</td><td colspan="2">Protocol</td><td colspan="4">Header Checksum</td></tr>
<tr><td colspan="8">Source Address</td></tr>
<tr><td colspan="8">Destination Address</td></tr>
<tr><td colspan="8">Options + Padding</td></tr>
</table>

- Note the mutable and immutable fields of an IPv4 header

For interest only – understand difference between mutable and immutable but no need to know what field is which type.

•Note that Protocol = UDP / TCP / ICMP / … / ESP / AH

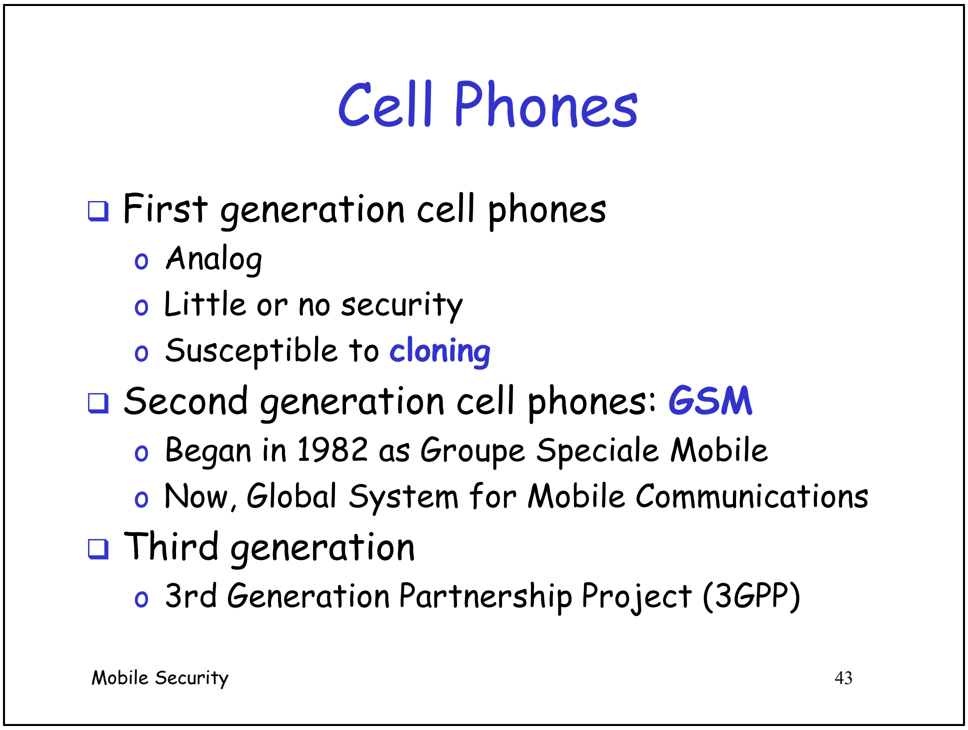
•The IPv4 header length is usually constant considering that there are no options.

On the other hand, the options has a bounded length.

Why Does AH Exist?

- No confidentiality
- AH authenticates **immutable fields** in IP header only
  - TTL, for example, must change
- ESP can provide both confidentiality and integrity (not of the IP header)

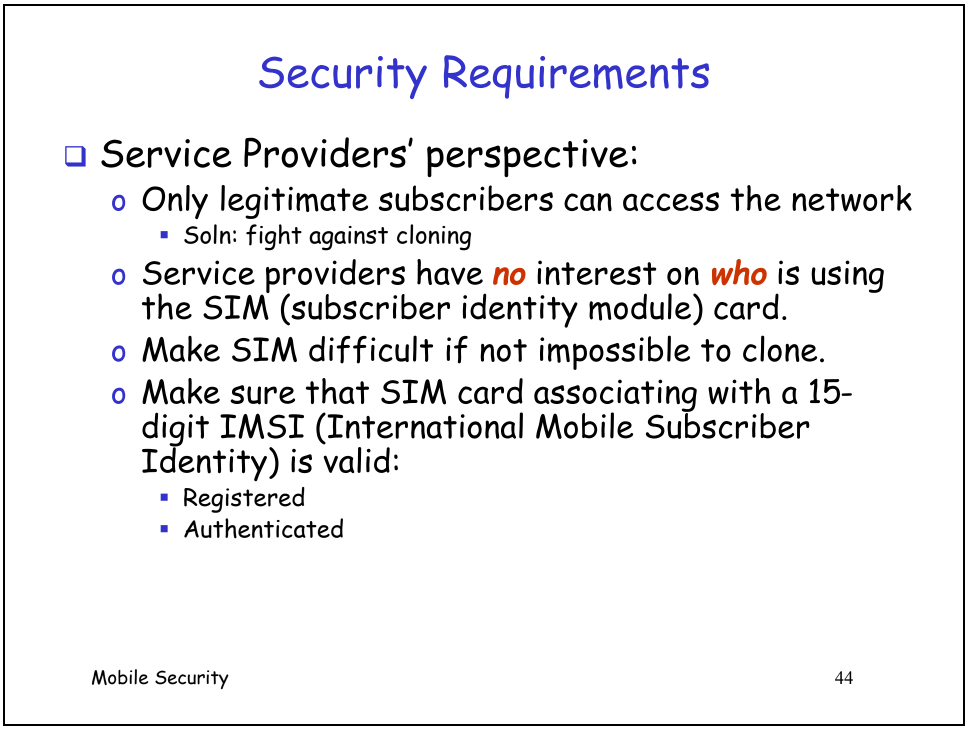
Mobile Network Security



Cell Phones

- First generation cell phones
  - Analog
  - Little or no security
  - Susceptible to **cloning**
- Second generation cell phones: ***GSM***
  - Began in 1982 as Groupe Speciale Mobile
  - Now, Global System for Mobile Communications
- Third generation
  - 3rd Generation Partnership Project (3GPP)

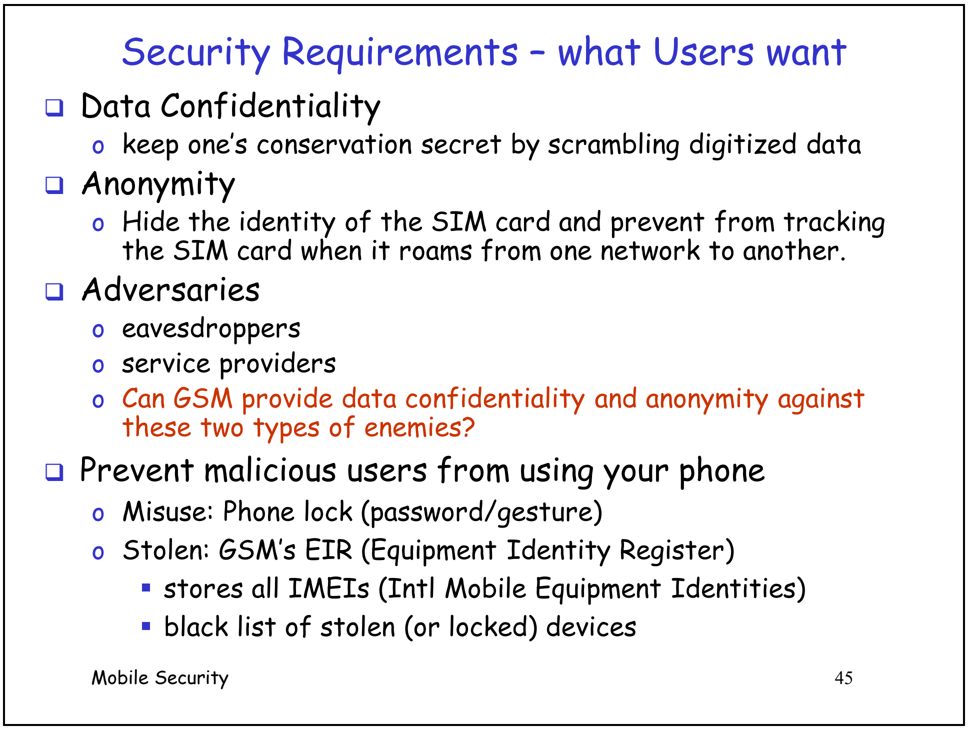
Slides 43-46 for interest



Security Requirements

- ❑ Service Providers' perspective:
  - Only legitimate subscribers can access the network
    - Soln: fight against cloning
  - Service providers have ***no*** interest on ***who*** is using the SIM (subscriber identity module) card.
  - Make SIM difficult if not impossible to clone.
  - Make sure that SIM card associating with a 15-digit IMSI (International Mobile Subscriber Identity) is valid:
    - Registered
    - Authenticated

We are considering security from different perspective than the user.



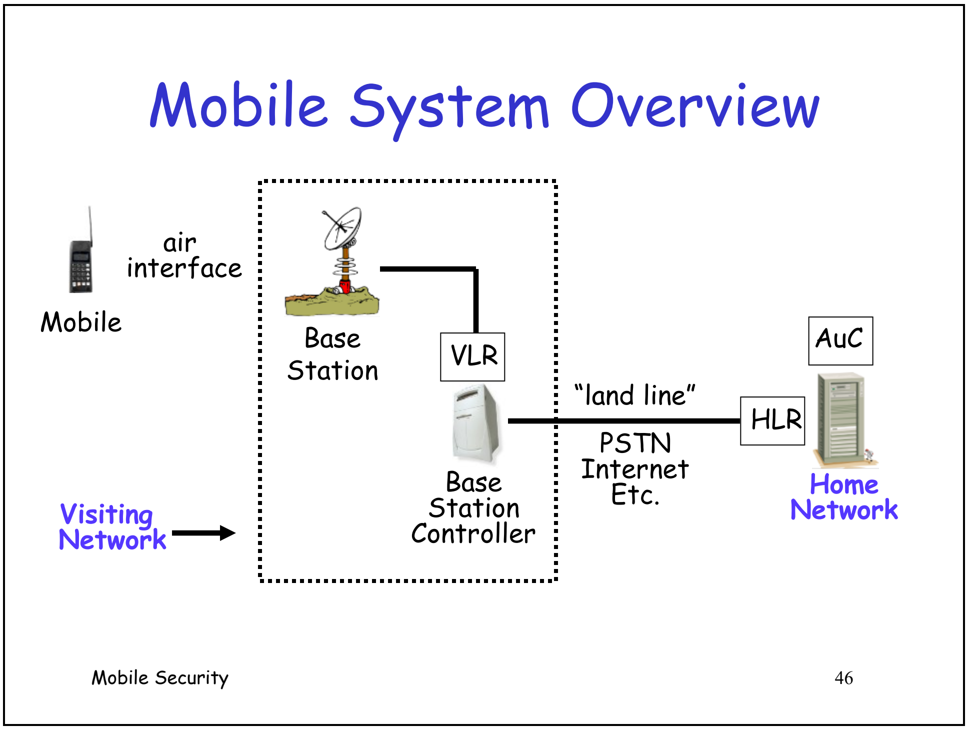
Security Requirements – what Users want

- ❑ Data Confidentiality
  - o keep one's conservation secret by scrambling digitized data
- ❑ Anonymity
  - o Hide the identity of the SIM card and prevent from tracking the SIM card when it roams from one network to another.
- ❑ Adversaries
  - o eavesdroppers
  - o service providers
  - o Can GSM provide data confidentiality and anonymity against these two types of enemies?
- ❑ Prevent malicious users from using your phone
  - o Misuse: Phone lock (password/gesture)
  - o Stolen: GSM's EIR (Equipment Identity Register)
    - ▪ stores all IMEIs (Intl Mobile Equipment Identities)
    - ▪ black list of stolen (or locked) devices

Lock not perfect: Grabbing phones, steal while unlocked (pattern locks look at smudges)

Dangers of data on devices, unlock password, backed up to cloud….

Apple unlock (clone phone, try PIN until locked out, reflash device)



Mobile System Overview

Mobile

air interface

Base Station

VLR

Base Station Controller

Visiting Network

"land line"

PSTN Internet Etc.

HLR

AuC

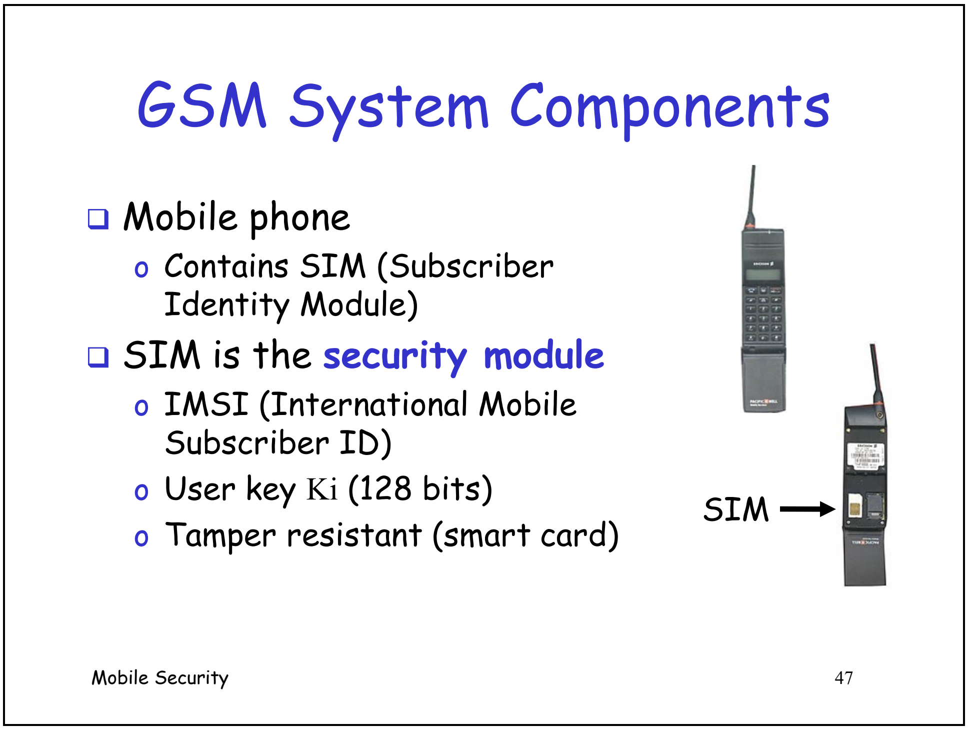
Home Network

Slide 47-49 you do not need to study but need to know well enough so you know who the parties are when we get to the security aspects.

VLR – visiting location register

HLR – home location register

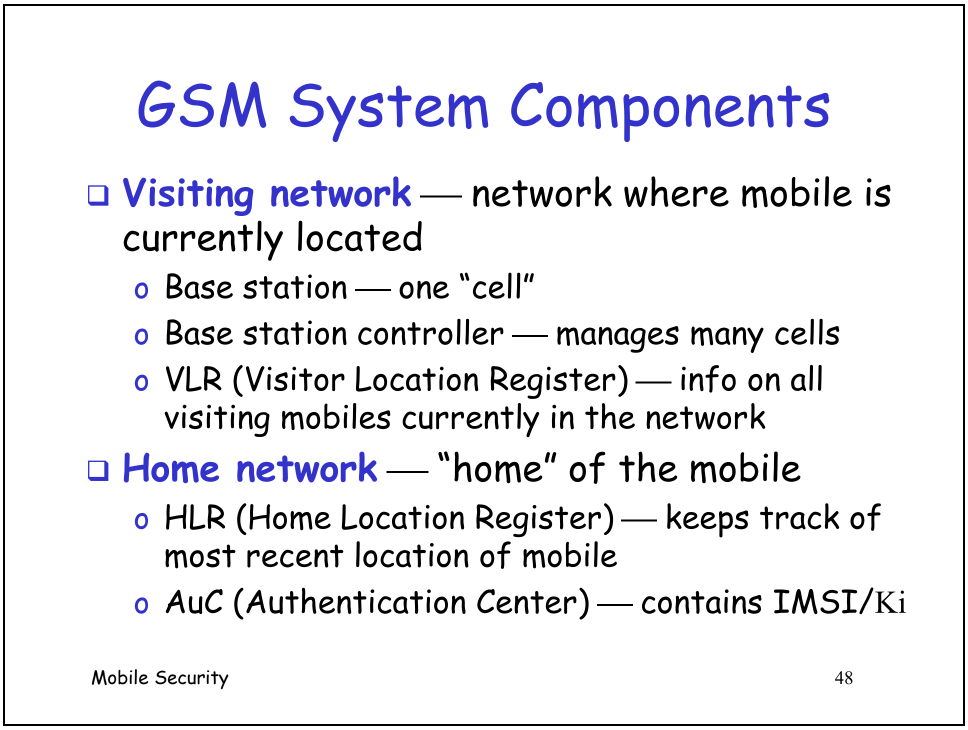
AuC – Authentication Centre



GSM System Components

- ❑ Mobile phone
  - o Contains SIM (Subscriber Identity Module)
- ❑ SIM is the **security module**
  - o IMSI (International Mobile Subscriber ID)
  - o User key Ki (128 bits)
  - o Tamper resistant (smart card)

Need to be able to give overview of security in mobile systems, and compare 2G and 3G security

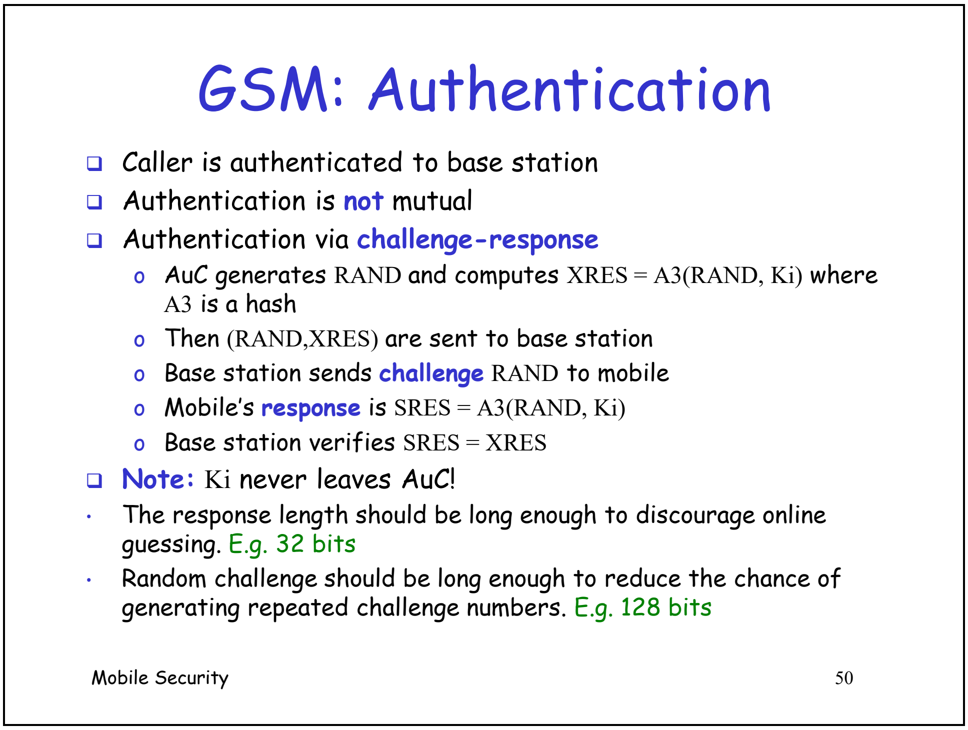


GSM System Components

- □ **Visiting network** — network where mobile is currently located
  - o Base station — one "cell"
  - o Base station controller — manages many cells
  - o VLR (Visitor Location Register) — info on all visiting mobiles currently in the network
- □ **Home network** — "home" of the mobile
  - o HLR (Home Location Register) — keeps track of most recent location of mobile
  - o AuC (Authentication Center) — contains IMSI/Ki

GSM: Anonymity

- ❑ IMSI used to initially identify caller
- ❑ Then TMSI (Temporary Mobile Subscriber ID) used
- ❑ TMSI changed frequently
- ❑ TMSI's encrypted when sent
- ❑ Not a strong form of anonymity
- ❑ But probably sufficient for most uses



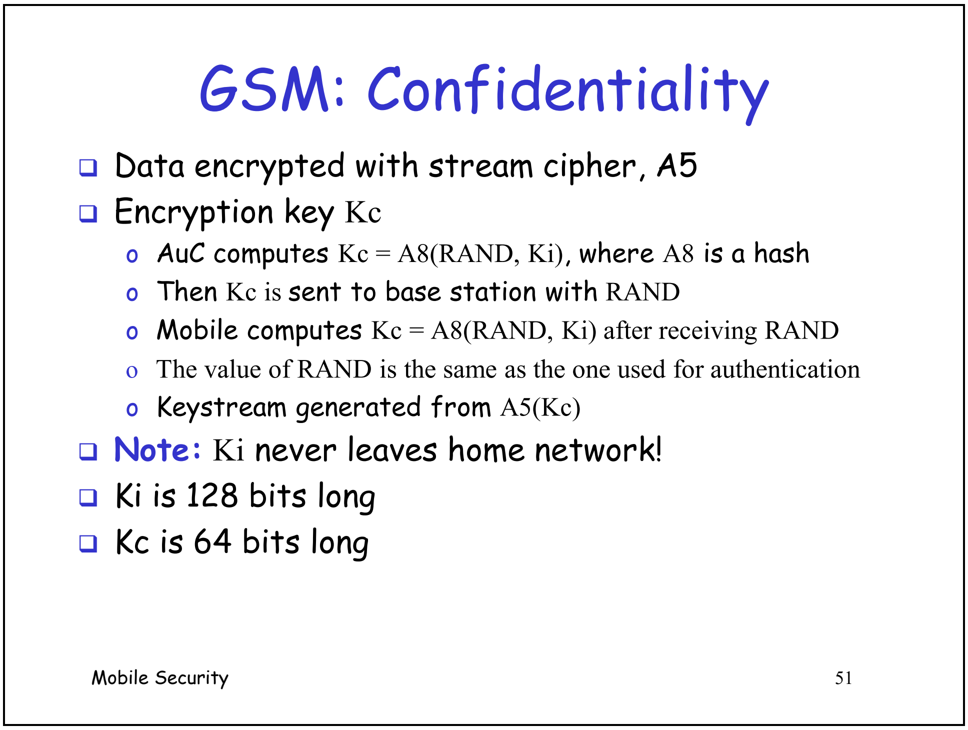
GSM: Authentication

- Caller is authenticated to base station
- Authentication is **not** mutual
- Authentication via **challenge-response**
  - AuC generates RAND and computes XRES = A3(RAND, Ki) where A3 is a hash
  - Then (RAND,XRES) are sent to base station
  - Base station sends **challenge** RAND to mobile
  - Mobile's **response** is SRES = A3(RAND, Ki)
  - Base station verifies SRES = XRES
- **Note:** Ki never leaves AuC!
- The response length should be long enough to discourage online guessing. E.g. 32 bits
- Random challenge should be long enough to reduce the chance of generating repeated challenge numbers. E.g. 128 bits

The rest of the slides on GSM(2G) and (3G) security you need to study. You do not need to remember algorithm names, key lengths, etc.

I expect that you would be able to give a basic summary of GSM security services and how GSM does authentication and confidentiality, e.g. SRES,RAND,Kc

I expect that you can do the same for 3G and that you can compare the two and say where 3G improves on 2G.



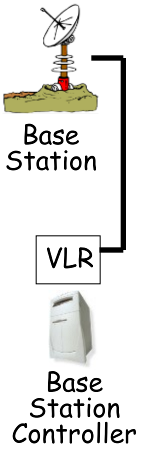
GSM: Confidentiality

- Data encrypted with stream cipher, A5
- Encryption key Kc
  - AuC computes Kc = A8(RAND, Ki), where A8 is a hash
  - Then Kc is sent to base station with RAND
  - Mobile computes Kc = A8(RAND, Ki) after receiving RAND
  - The value of RAND is the same as the one used for authentication
  - Keystream generated from A5(Kc)
- **Note:** Ki never leaves home network!
- Ki is 128 bits long
- Kc is 64 bits long

Triplet Kc, XRES, RAND

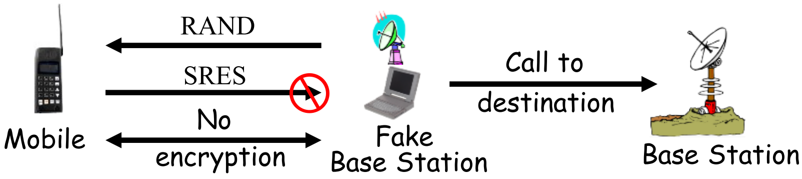
GSM Insecurity (1)

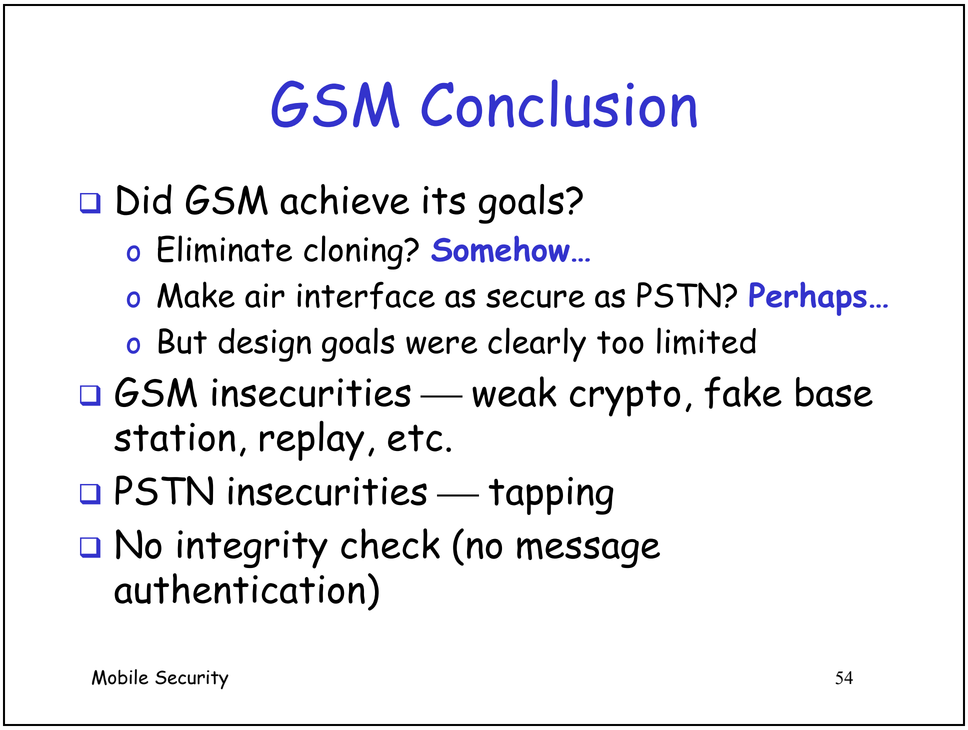
- Hash used in A3/A8:
  - Broken after 160,000 chosen plaintexts
  - With SIM, can get Ki in 2 to 10 hours
- Encryption between mobile and base station but **no encryption** from base station to base station controller
  - When transmitted over microwave link…
- Encryption algorithm A5/1
  - Broken with 2 seconds of known plaintext



GSM Insecurity (2)

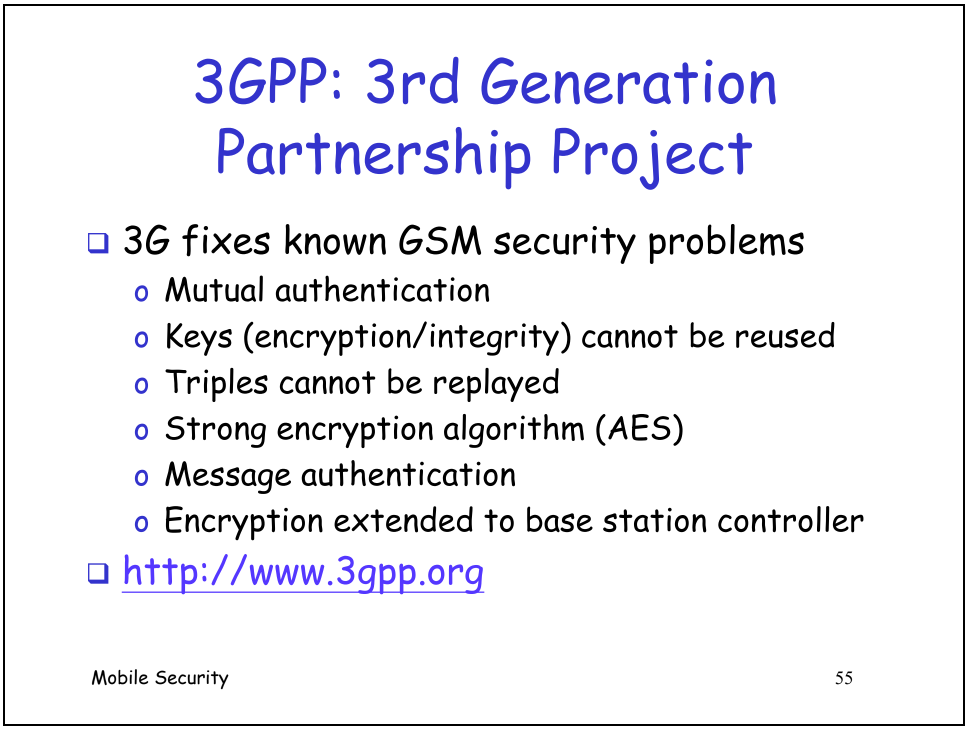
- **Fake base station** exploits two flaws
  - Encryption not automatic
  - Base station not authenticated





GSM Conclusion

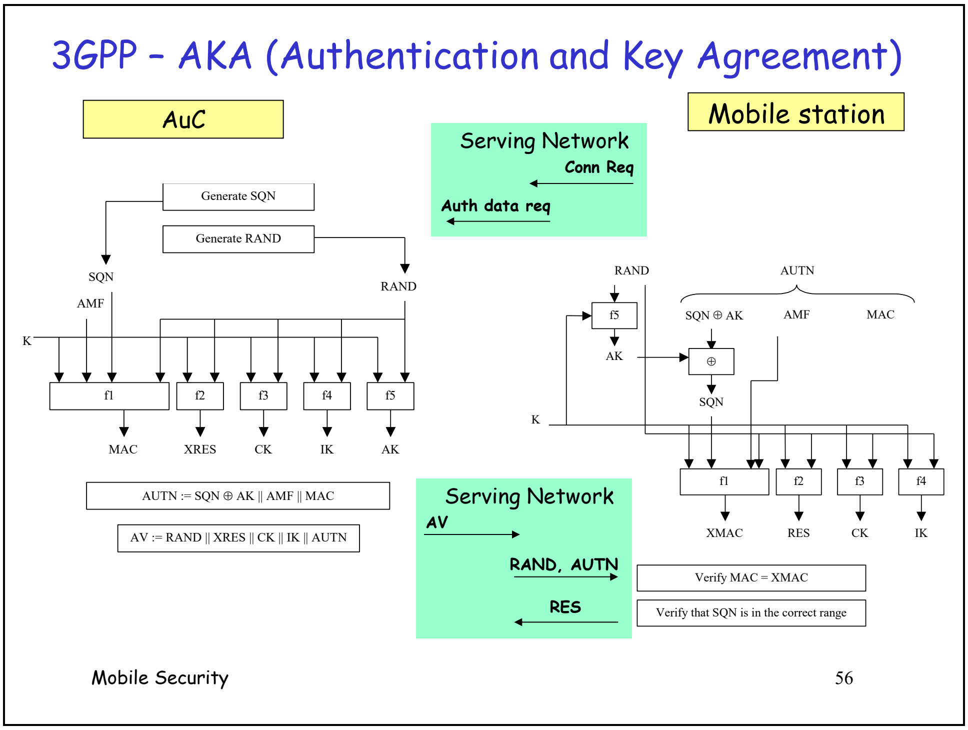
- Did GSM achieve its goals?
  - Eliminate cloning? **Somehow…**
  - Make air interface as secure as PSTN? **Perhaps…**
  - But design goals were clearly too limited
- GSM insecurities — weak crypto, fake base station, replay, etc.
- PSTN insecurities — tapping
- No integrity check (no message authentication)



3GPP: 3rd Generation Partnership Project

- ❑ 3G fixes known GSM security problems
  - o Mutual authentication
  - o Keys (encryption/integrity) cannot be reused
  - o Triples cannot be replayed
  - o Strong encryption algorithm (AES)
  - o Message authentication
  - o Encryption extended to base station controller
- ❑ http://www.3gpp.org

Triplet Kc, XRES, RAND



3GPP – AKA (Authentication and Key Agreement)

AuC

Serving Network

Mobile station

Conn Req

Auth data req

Generate SQN

Generate RAND

SQN

AMF

RAND

K

f1 f2 f3 f4 f5

MAC XRES CK IK AK

AUTN := SQN ⊕ AK || AMF || MAC

AV := RAND || XRES || CK || IK || AUTN

Serving Network

AV

RAND, AUTN

RES

RAND

AUTN

f5

AK

SQN ⊕ AK

AMF

MAC

⊕

SQN

K

f1 f2 f3 f4

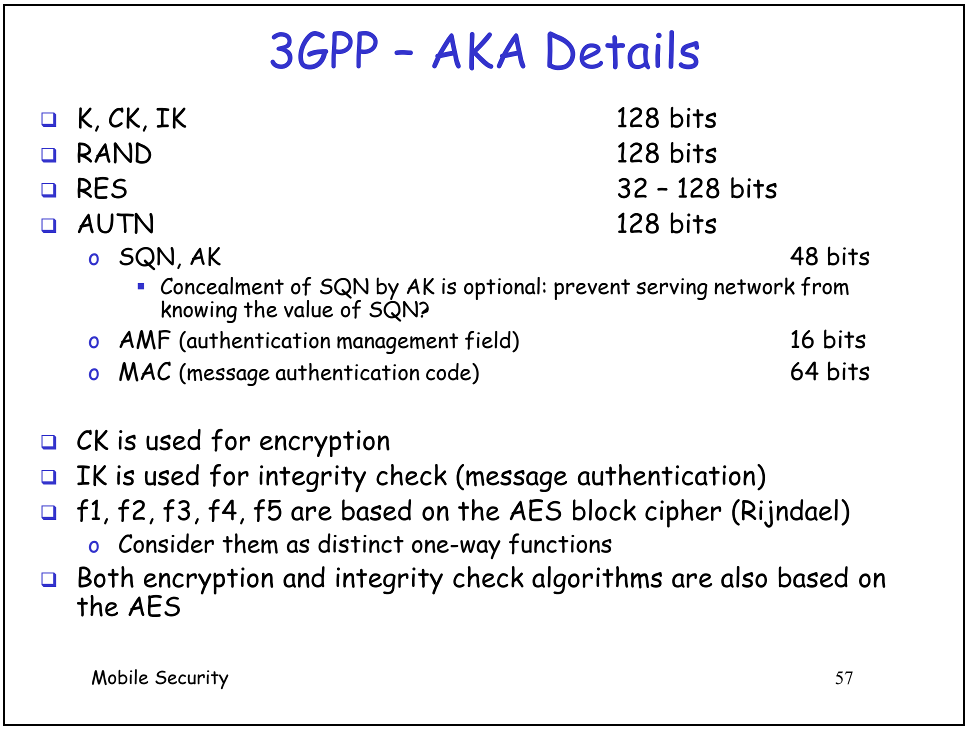
XMAC RES CK IK

Verify MAC = XMAC

Verify that SQN is in the correct range

All of this is function of the ky

AMF (authentication management field)

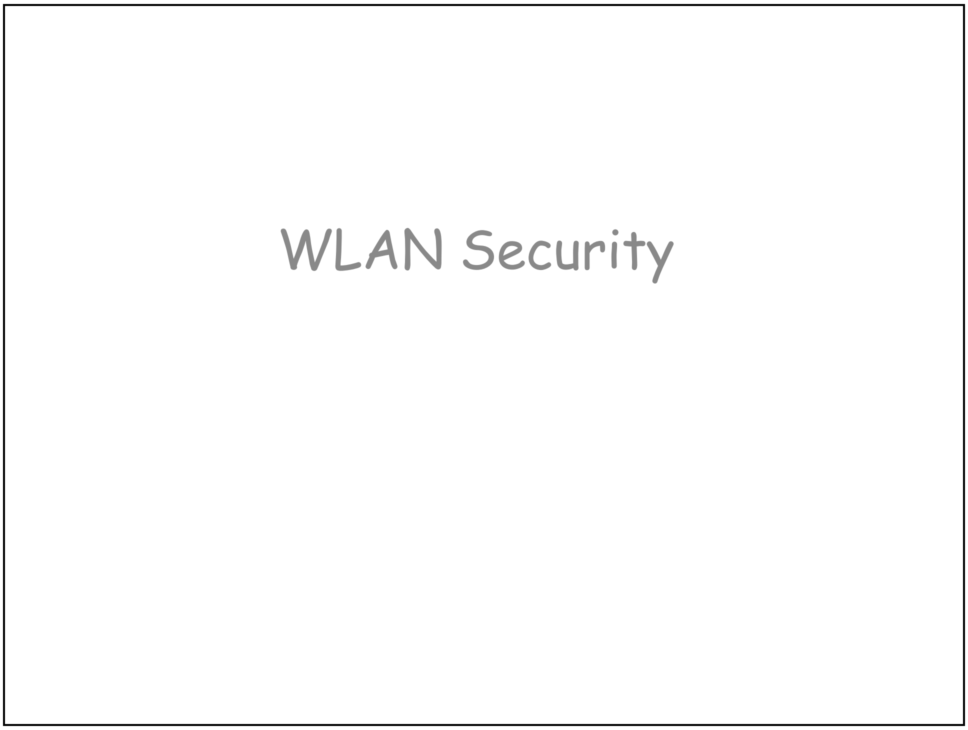


3GPP – AKA Details

K, CK, IK	128 bits
RAND	128 bits
RES	32 – 128 bits
AUTN	128 bits
o SQN, AK	48 bits
▪ Concealment of SQN by AK is optional: prevent serving network from knowing the value of SQN?	
o AMF (authentication management field)	16 bits
o MAC (message authentication code)	64 bits

- CK is used for encryption
- IK is used for integrity check (message authentication)
- f1, f2, f3, f4, f5 are based on the AES block cipher (Rijndael)
  - Consider them as distinct one-way functions
- Both encryption and integrity check algorithms are also based on the AES

For interest



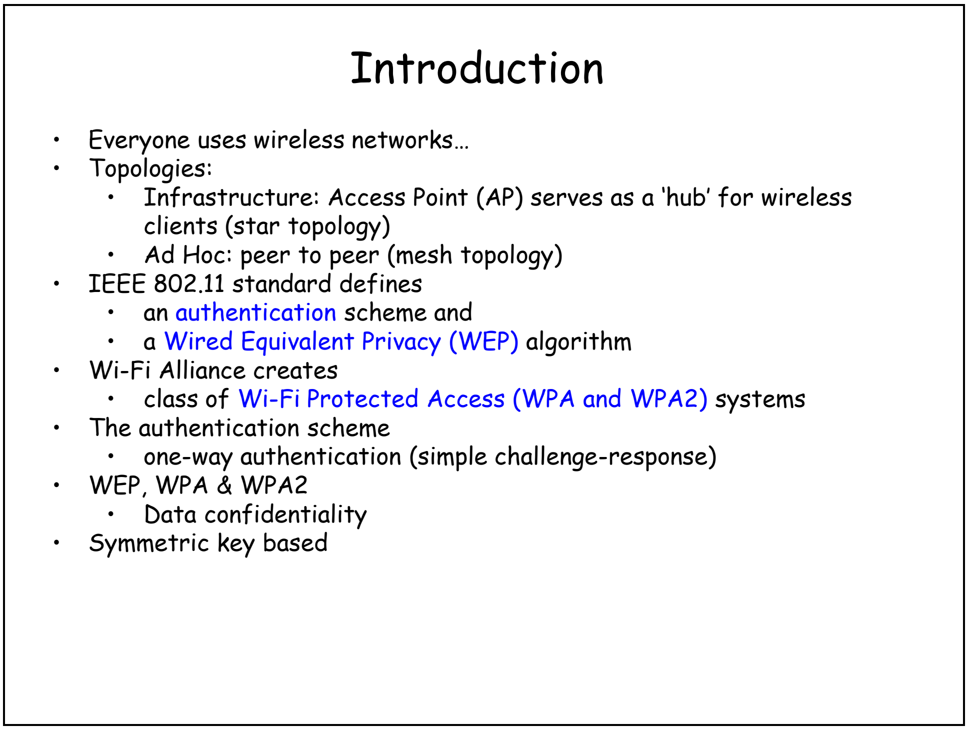
WLAN Security

For WLAN security all I want you to study is the basic differences between WEP, WPA and WPA2. So read through the slides and make your own short summary of the different modes and their differences.

For example, WEP has a long term key, RC4, bad integrity measures and was pretty insecure.

WPA mad an improvement on key reuse (it had key distributed by a TTP) but still RC4

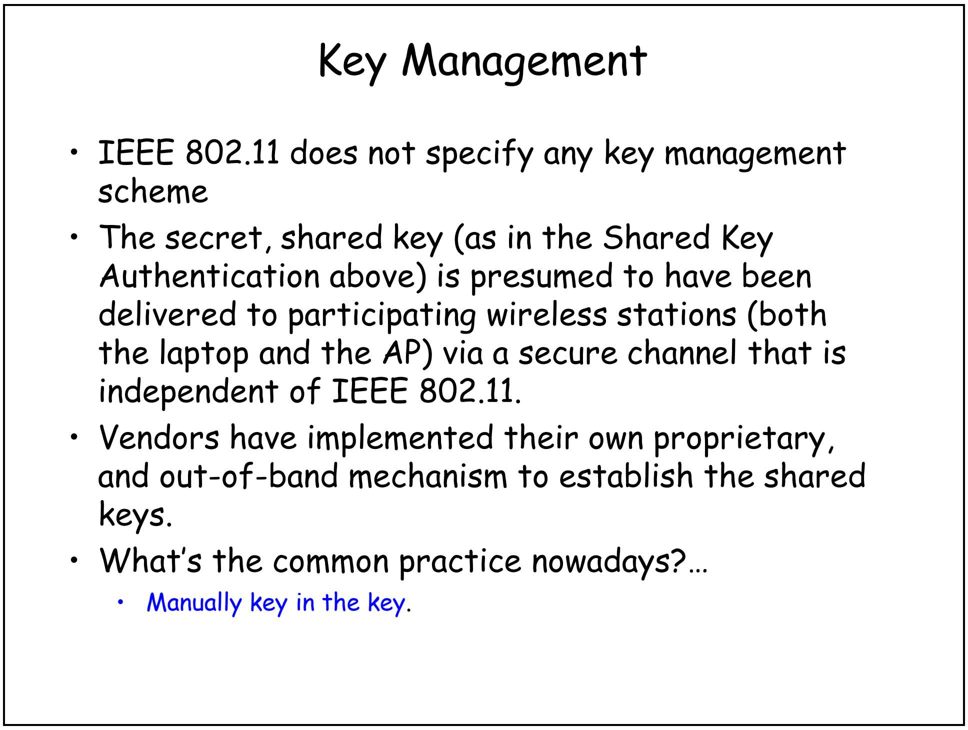
WPA2 started using AES.



Introduction

- Everyone uses wireless networks…
- Topologies:
  - Infrastructure: Access Point (AP) serves as a ‘hub’ for wireless clients (star topology)
  - Ad Hoc: peer to peer (mesh topology)
- IEEE 802.11 standard defines
  - an authentication scheme and
  - a Wired Equivalent Privacy (WEP) algorithm
- Wi-Fi Alliance creates
  - class of Wi-Fi Protected Access (WPA and WPA2) systems
- The authentication scheme
  - one-way authentication (simple challenge-response)
- WEP, WPA & WPA2
  - Data confidentiality
- Symmetric key based

WPA WiFi Protected Access

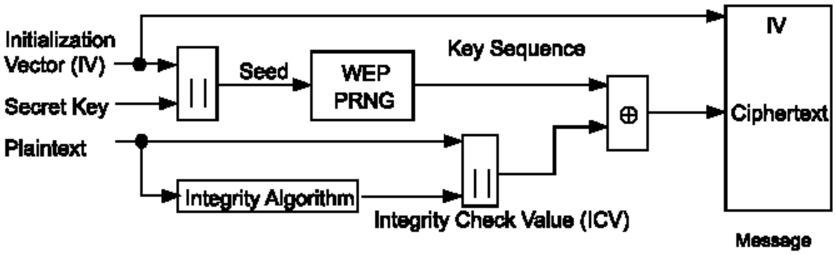


Key Management

- IEEE 802.11 does not specify any key management scheme
- The secret, shared key (as in the Shared Key Authentication above) is presumed to have been delivered to participating wireless stations (both the laptop and the AP) via a secure channel that is independent of IEEE 802.11.
- Vendors have implemented their own proprietary, and out-of-band mechanism to establish the shared keys.
- What's the common practice nowadays?…
  - Manually key in the key.

WEP

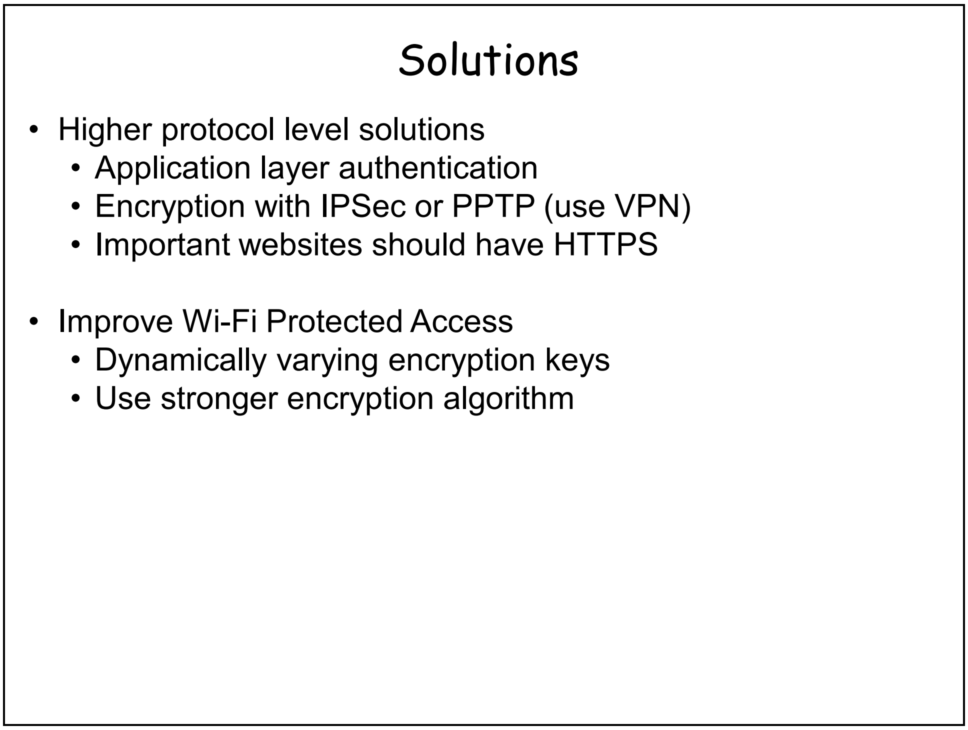
- WEP encipherment block diagram



- Secret Key: 40 bits or 104 bits
  - Distributed to communicating entities (wireless stations and access points) via external key management service (e.g. manually key in)
- Integrity Algorithm
  - CRC-32
- WEP PRNG
  - RC4
  - Initialized by Seed
  - Outputs a long binary stream called Key Sequence

WEP Weaknesses

- 2001: WEP was broken. Attacking Principles
  - The first byte of an encrypted message is always equal to 0xAA. Hence the first byte of key sequence is always obtainable.
  - For some special pattern of the 24-bit IV, one can deduce one byte of the secret key at one time. When enough IVs and ciphertexts have been collected, all bytes of the secret key can be obtained.
- Several other weaknesses have been identified since the publication of the algorithm.
  - Static key (difficult to update), weak linear (CRC) integrity
- Open-source cracking software is now available on the Internet.
  - AirSnort (http://airsnort.shmoo.com)
  - WEPcrack (http://wepcrack.sourceforge.net/)
  - Aircrack (http://aircrack-ng.org/doku.php)



Solutions

- Higher protocol level solutions
  - Application layer authentication
  - Encryption with IPSec or PPTP (use VPN)
  - Important websites should have HTTPS

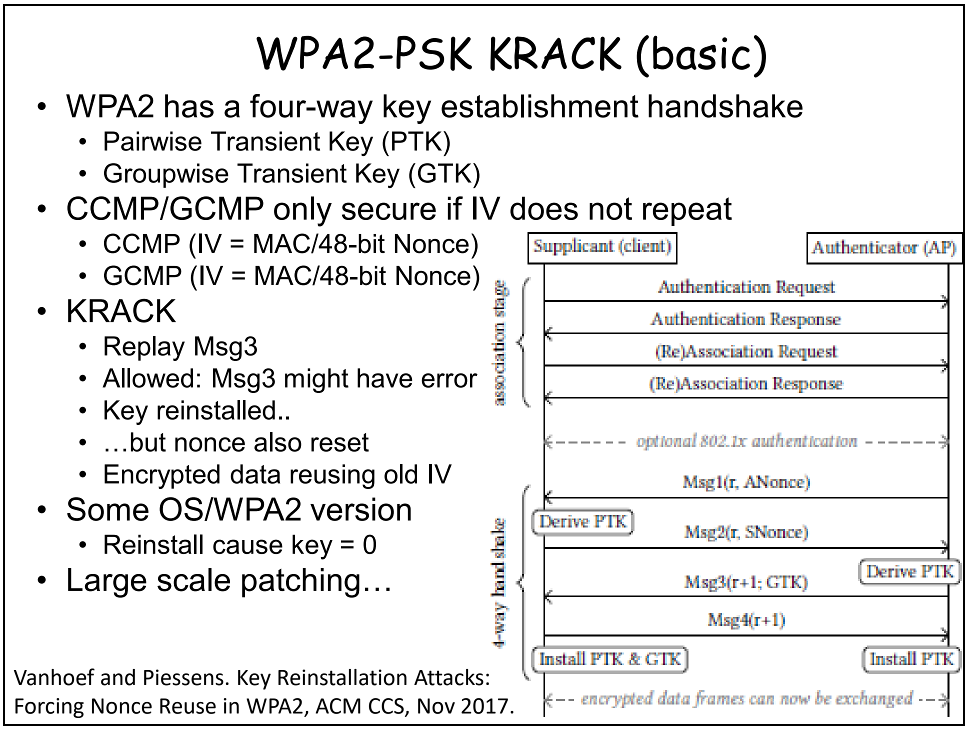
- Improve Wi-Fi Protected Access
  - Dynamically varying encryption keys
  - Use stronger encryption algorithm

WPA/WPA2

- Created by the Wi-Fi Alliance; supersedes WEP in 2003

- WPA (go for very regular key changes)
    - 802.1x with Extensible Authentication Protocol (EAP) for authentication
        - Allows for session keys
    - Temporal Key Integrity Protocol (TKIP) for encryption
        - Dynamically generated 128-bit key (updated ±10,000 packets)
        - Still uses RC4 stream cipher (WPA)
    - Proprietary Michael integrity check algorithm

- WPA2 (go for stronger cryptography and session keys)
    - CCMP (IEEE 802.11i)
        - AES (still operating like a stream cipher in Counter Mode)
        - CBC-MAC for integrity
    - GCMP (802.11ad)
        - AES in GCM mode (Galios/Counter Mode)
        - GHASH for integrity
    - Allows alternative to have a password shared between an AP and a user (Pre-shared key "WPA2-PSK" mode)



WPA2-PSK KRACK (basic)

- WPA2 has a four-way key establishment handshake
  - Pairwise Transient Key (PTK)
  - Groupwise Transient Key (GTK)
- CCMP/GCMP only secure if IV does not repeat
  - CCMP (IV = MAC/48-bit Nonce)
  - GCMP (IV = MAC/48-bit Nonce)
- KRACK
  - Replay Msg3
  - Allowed: Msg3 might have error
  - Key reinstalled..
  - …but nonce also reset
  - Encrypted data reusing old IV
- Some OS/WPA2 version
  - Reinstall cause key = 0
- Large scale patching…

Vanhoef and Piessens. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2, ACM CCS, Nov 2017.

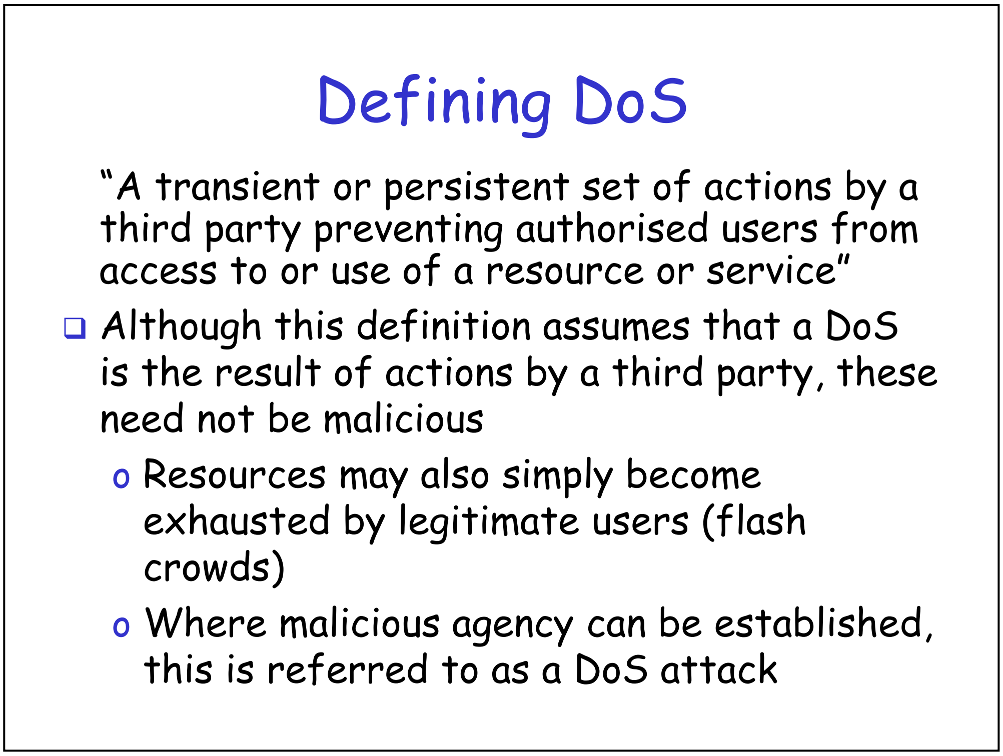
Once again, we can look beyond the protocol to have security (even though protocol is secure, some implementation was not).

KRACK – Key reinstallation attacks

For more read
https://www.krackattacks.com/

October 2017

DoS and DDoS



Defining DoS

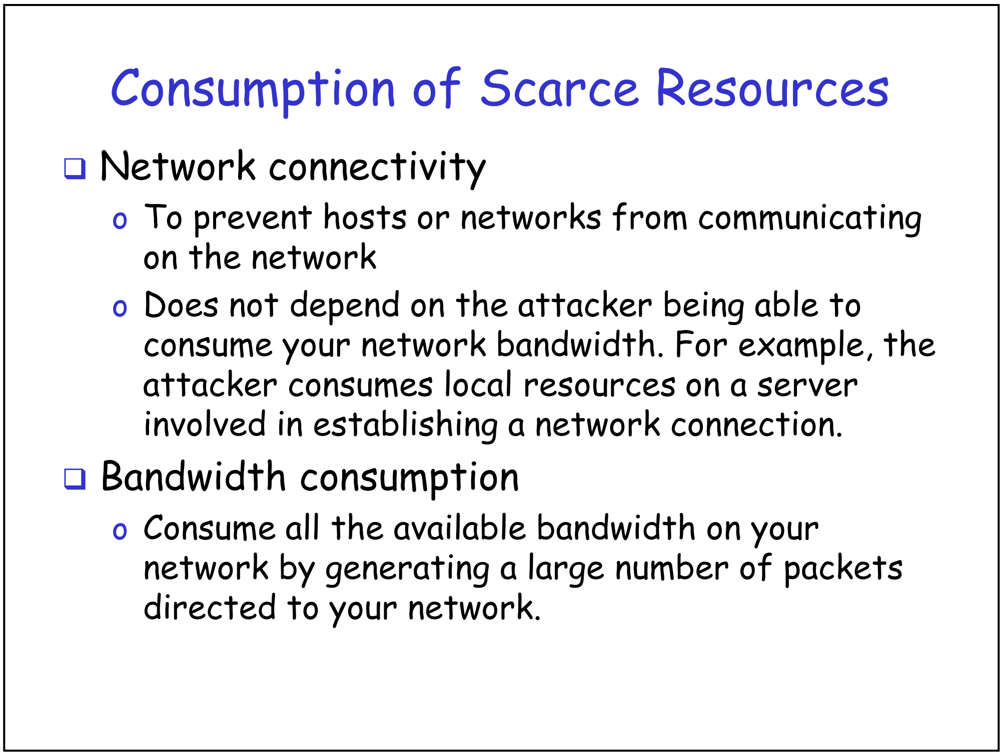
"A transient or persistent set of actions by a third party preventing authorised users from access to or use of a resource or service"

- Although this definition assumes that a DoS is the result of actions by a third party, these need not be malicious
  - Resources may also simply become exhausted by legitimate users (flash crowds)
  - Where malicious agency can be established, this is referred to as a DoS attack

What security service does DoS influence? – availability

Sino Weibo – famous singer introduced his new girlfriend and it generated some much traffic the servers crashed.

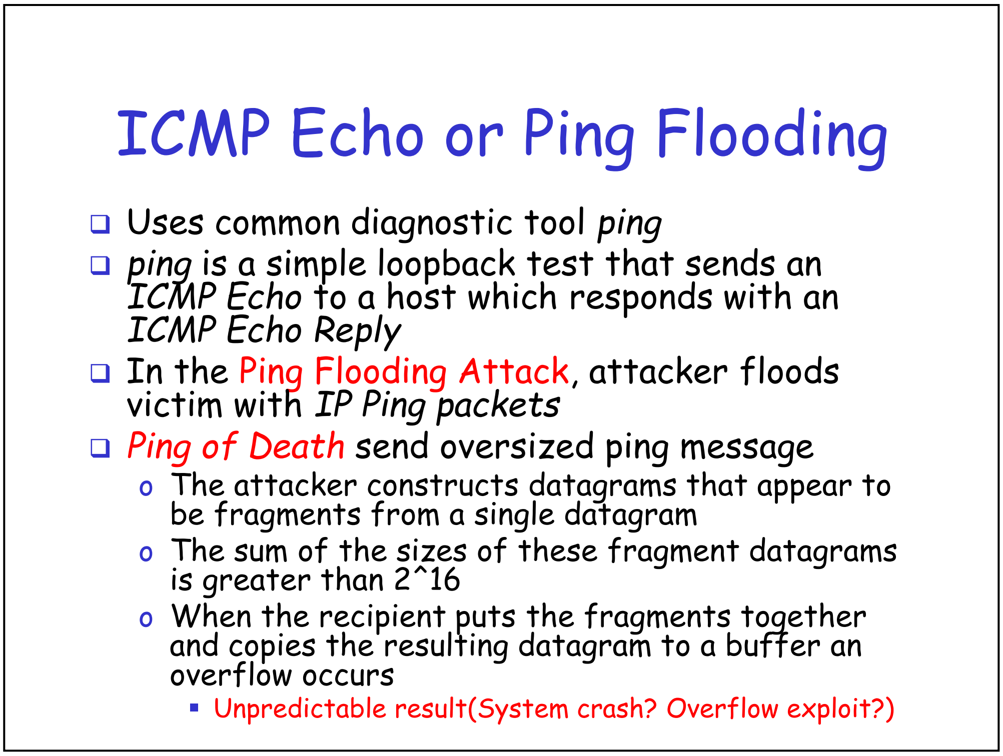
https://www.cnet.com/tech/services-and-software/chinese-internet-users-sent-weibo-crashing-for-hours/



Consumption of Scarce Resources

- ❑ Network connectivity
  - o To prevent hosts or networks from communicating on the network
  - o Does not depend on the attacker being able to consume your network bandwidth. For example, the attacker consumes local resources on a server involved in establishing a network connection.
- ❑ Bandwidth consumption
  - o Consume all the available bandwidth on your network by generating a large number of packets directed to your network.

Right so you two main resources that jump into you mind when talking DoS



ICMP Echo or Ping Flooding

- Uses common diagnostic tool *ping*
- *ping* is a simple loopback test that sends an *ICMP Echo* to a host which responds with an *ICMP Echo Reply*
- In the Ping Flooding Attack, attacker floods victim with *IP Ping packets*
- *Ping of Death* send oversized ping message
  - The attacker constructs datagrams that appear to be fragments from a single datagram
  - The sum of the sizes of these fragment datagrams is greater than 2^16
  - When the recipient puts the fragments together and copies the resulting datagram to a buffer an overflow occurs
    - Unpredictable result(System crash? Overflow exploit?)

Interest only

Again potentially with IP spoofing

ICMP (Internet Control Message Protocol) – error handling and control messages on the internet.

ICMP is a subset of the TCP/IP suite of protocols that transmits error and control messages between systems. Two specific instances of the ICMP are the ICMP ECHO_REQUEST and ICMP ECHO_RESPONSE datagrams. These two instances can be used by a local host to determine whether a remote system is reachable via the network; this is commonly achieved using the "ping" command.

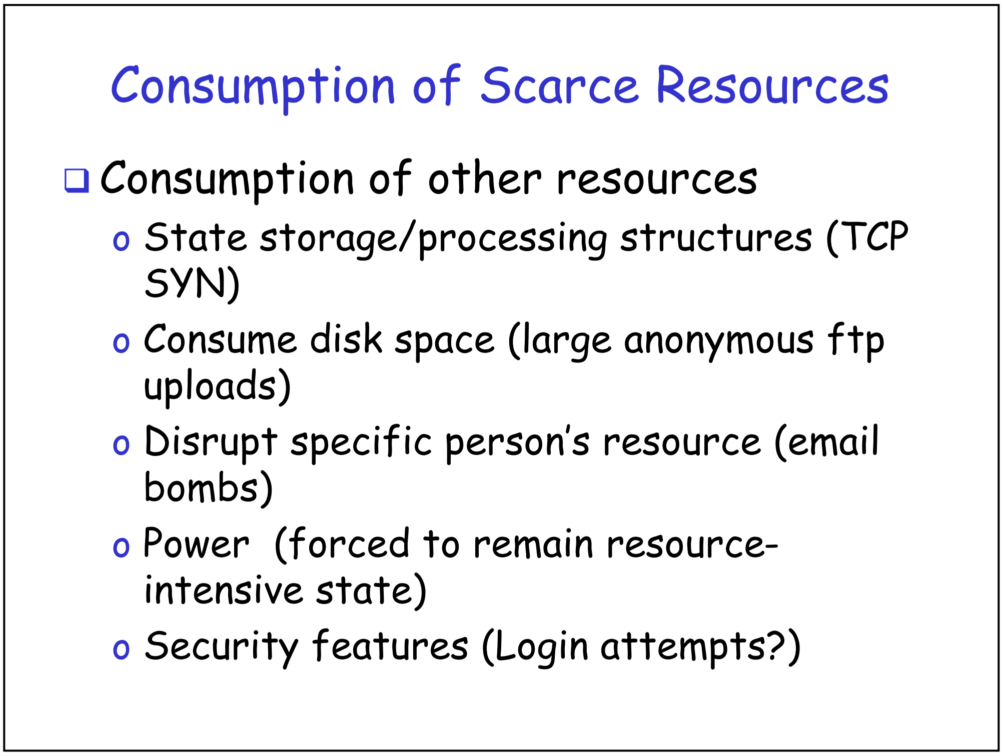
Go through the slide…

Ping of death?

It is known that some systems will react in an unpredictable fashion when receiving oversized IP packets. Reports indicate a range of reactions including crashing, freezing, and rebooting.

In particular, the reports received by the CERT Coordination Center indicate that Internet Control Message Protocol (ICMP) packets issued via the "ping" command have been used to trigger this behavior.

Discussion has centered around the use of the "ping" command to construct oversized ICMP datagrams (which are encapsulated within an IP packet). Many ping implementations by default send ICMP datagrams consisting only of the 8 octets of ICMP header information but allow the user to specify a larger packet size if desired.



Consumption of Scarce Resources

- Consumption of other resources
  - State storage/processing structures (TCP SYN)
  - Consume disk space (large anonymous ftp uploads)
  - Disrupt specific person's resource (email bombs)
  - Power (forced to remain resource-intensive state)
  - Security features (Login attempts?)

For example, in many systems, a limited number of data structures are available to hold process information (process identifiers, process table entries, process slots, etc.). An intruder may be able to consume these data structures by writing a simple program or script that does nothing but repeatedly create copies of itself.

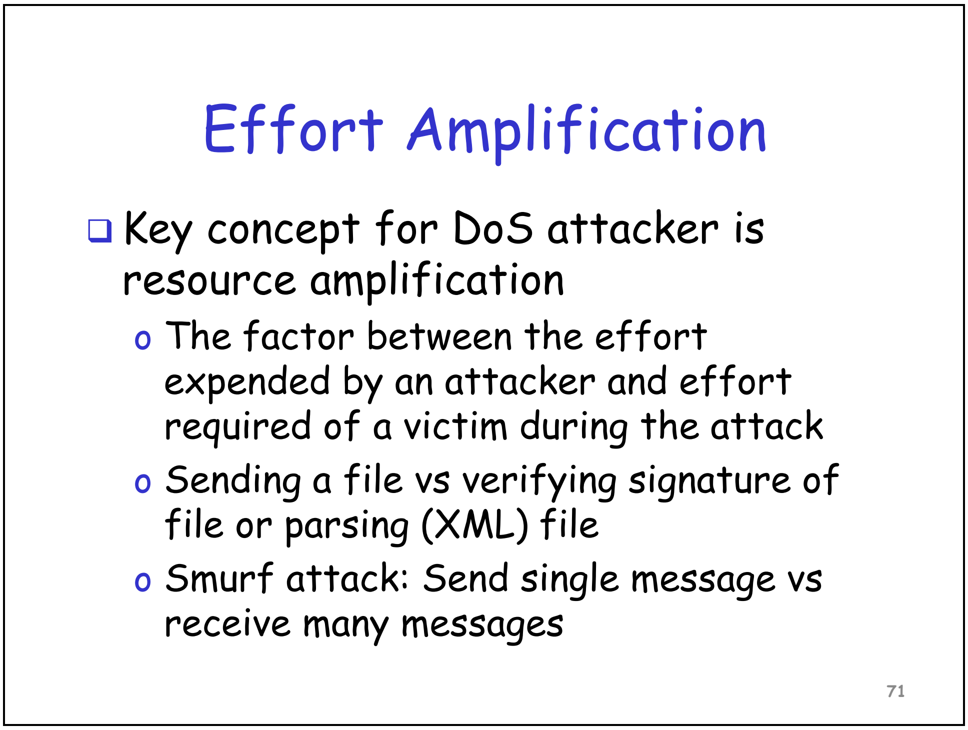
An intruder may also attempt to consume disk space in other ways, including

generating excessive numbers of mail messages.

intentionally generating errors that must be logged placing files in anonymous ftp areas or network shares

In general, anything that allows data to be written to disk can be used to execute a denial-of-service attack if there are no bounds on the amount of data that can be written.

Also, many sites have schemes in place to "lockout" an account after a certain number of failed login attempts. A typical set up locks out an account after 3 or 5 failed login attempts. An intruder may be able to use this scheme to prevent legitimate users from logging in. In some cases, even the privileged accounts, such as root or administrator, may be subject to this type of attack.



Effort Amplification

- ❑ Key concept for DoS attacker is resource amplification
  - o The factor between the effort expended by an attacker and effort required of a victim during the attack
  - o Sending a file vs verifying signature of file or parsing (XML) file
  - o Smurf attack: Send single message vs receive many messages

71

Amplification

Otherwise it could just become a resource war issue. Can I generate more than you can receive?

DNS – looks at the Spamhaus incident you are looking at a DNS bases DoS attack that peaked at 300 Gbps… (2013)

NTP – Cloudflare 400 Gbps (2014)

Security protocols tend to be asymmetric (not asymetric crypto) – generally authentication one party does more than the other. This is being address forcing more of the workload onto the clients.

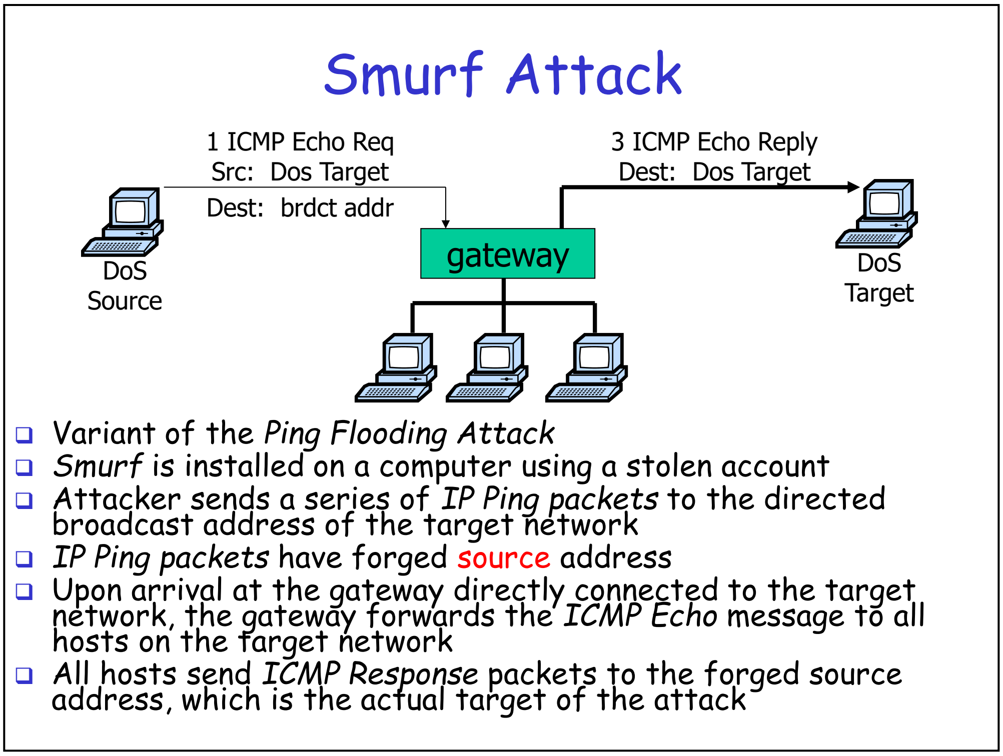
Other example XML (and layered protocols) are targets, e.g. for:

Jumbo payloads Large payloads designed to exhaust memory and CPU on the victim machine {these can be generated in a way that they do not have to be stored on the attacking system

Recursive elements Forcing recursive entity expansion or repeated processing

Large tags Large numbers of tags, or long names (may also cause buffer overflows)

Coercive parsing Creating messages known to require significant resources for parsing



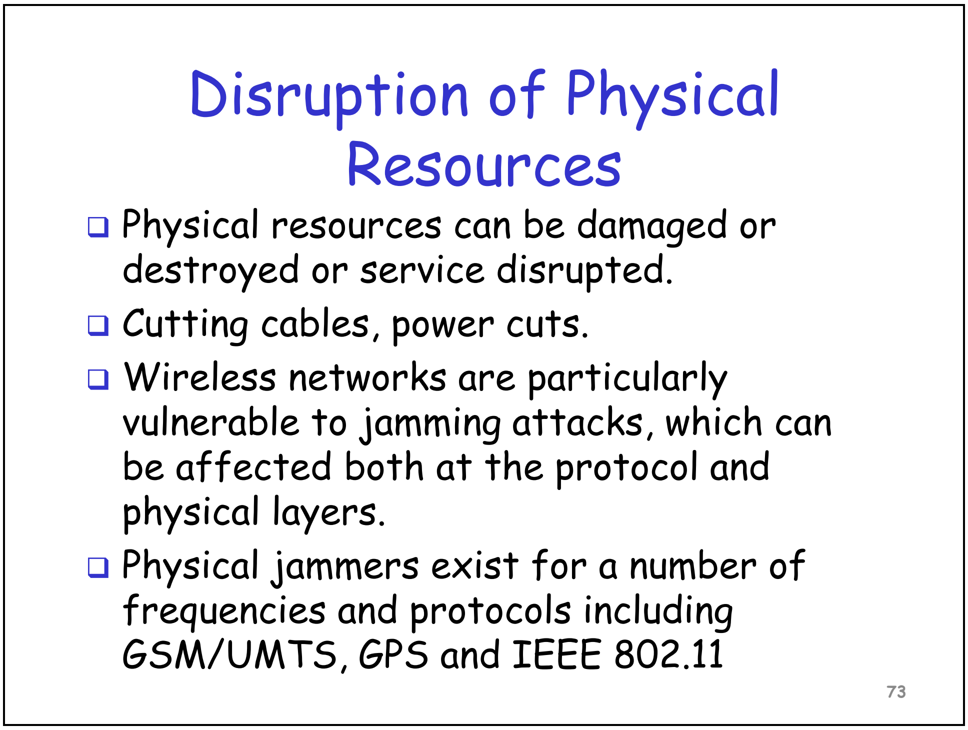
Smurf Attack

- Variant of the *Ping Flooding Attack*
- *Smurf* is installed on a computer using a stolen account
- Attacker sends a series of *IP Ping packets* to the directed broadcast address of the target network
- *IP Ping packets* have forged source address
- Upon arrival at the gateway directly connected to the target network, the gateway forwards the *ICMP Echo* message to all hosts on the target network
- All hosts send *ICMP Response* packets to the forged source address, which is the actual target of the attack

Interest only

What can you do against this?

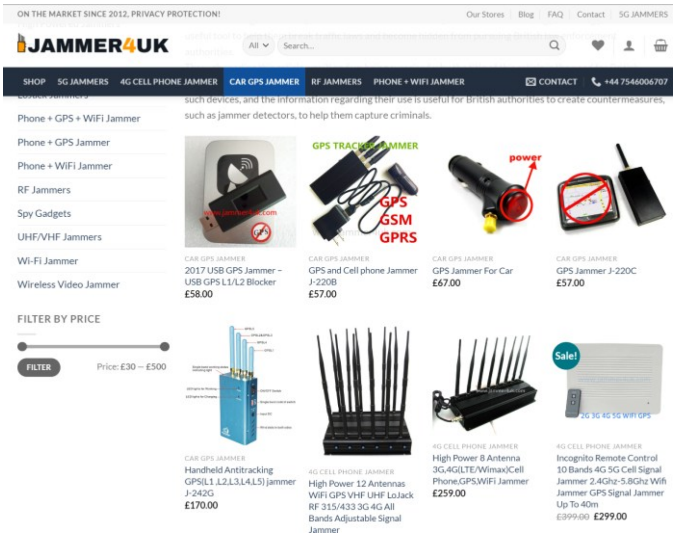
Disable IP-directed broadcast functionality on all routers.



Disruption of Physical Resources

- ❑ Physical resources can be damaged or destroyed or service disrupted.
- ❑ Cutting cables, power cuts.
- ❑ Wireless networks are particularly vulnerable to jamming attacks, which can be affected both at the protocol and physical layers.
- ❑ Physical jammers exist for a number of frequencies and protocols including GSM/UMTS, GPS and IEEE 802.11

Disruption of Physical



ON THE MARKET SINCE 2012, PRIVACY PROTECTION! Our Stores | Blog | FAQ | Contact | 5G JAMMERS

JAMMER4UK All Search...

SHOP 5G JAMMERS 4G CELL PHONE JAMMER CAR GPS JAMMER RF JAMMERS PHONE + WIFI JAMMER CONTACT +44 7546006707

such devices, and the information regarding their use is useful for British authorities to create countermeasures, such as jammer detectors, to help them capture criminals.

- Phone + GPS + WiFi Jammer
- Phone + GPS Jammer
- Phone + WiFi Jammer
- RF Jammers
- Spy Gadgets
- UHF/VHF Jammers
- Wi-Fi Jammer
- Wireless Video Jammer

FILTER BY PRICE

FILTER Price: £30 — £500

CAR GPS JAMMER
2017 USB GPS Jammer – USB GPS L1/L2 Blocker
£58.00

CAR GPS JAMMER
GPS and Cell phone Jammer J-220B
£57.00

CAR GPS JAMMER
GPS Jammer For Car
£67.00

CAR GPS JAMMER
GPS Jammer J-220C
£57.00

CAR GPS JAMMER
Handheld Antitracking GPS(L1 ,L2,L3,L4,L5) jammer J-242G
£170.00

4G CELL PHONE JAMMER
High Power 12 Antennas WiFi GPS VHF UHF LoJack RF 315/433 3G 4G All Bands Adjustable Signal Jammer

4G CELL PHONE JAMMER
High Power 8 Antenna 3G,4G(LTE/Wimax)Cell Phone,GPS,WiFi Jammer
£259.00

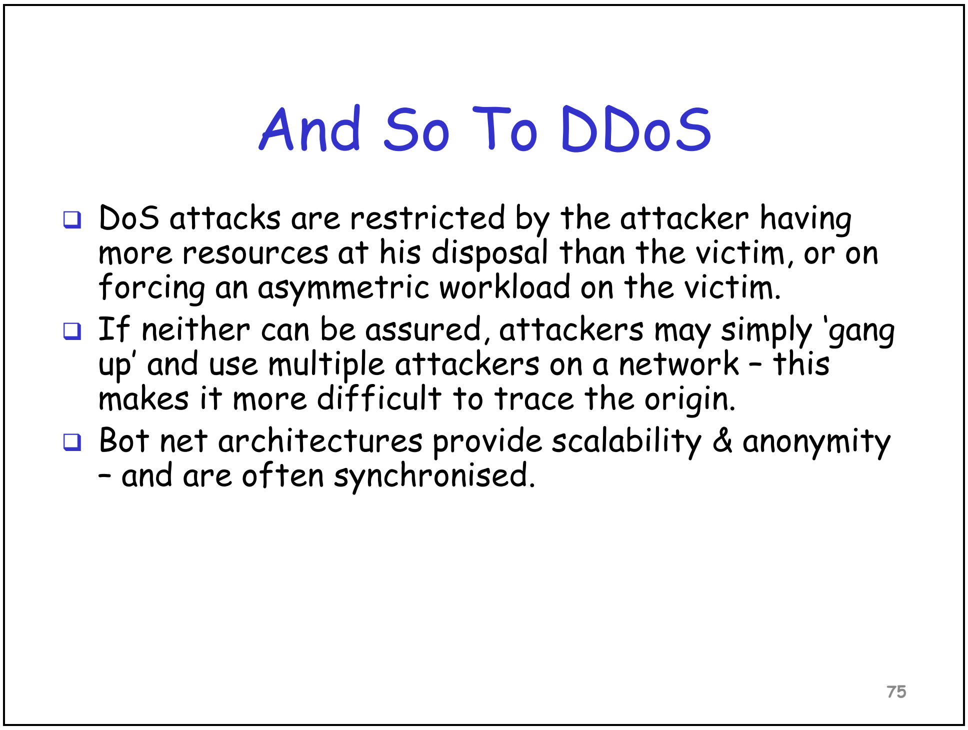
4G CELL PHONE JAMMER
Incognito Remote Control 10 Bands 4G 5G Cell Signal Jammer 2.4Ghz-5.8Ghz Wifi Jammer GPS Signal Jammer Up To 40m
~~£399.00~~ £299.00

Interest only

This is not an accident

Jammers are easy to buy, mobile, WiFi, GPS…

Newark airport personal privacy devices disrupt ground based augmentation system tracking airplanes



And So To DDoS

- DoS attacks are restricted by the attacker having more resources at his disposal than the victim, or on forcing an asymmetric workload on the victim.
- If neither can be assured, attackers may simply 'gang up' and use multiple attackers on a network – this makes it more difficult to trace the origin.
- Bot net architectures provide scalability & anonymity – and are often synchronised.

You need to know the general idea of what a DDoS attack is (see all remaining slides), and how it differs to DoS.

For attacks to be successful, these must be Synchronised

Botnets often have sophisticated C2 infrastructures

These are also often secured quite well to prevent other criminals from taking over an existing botnet

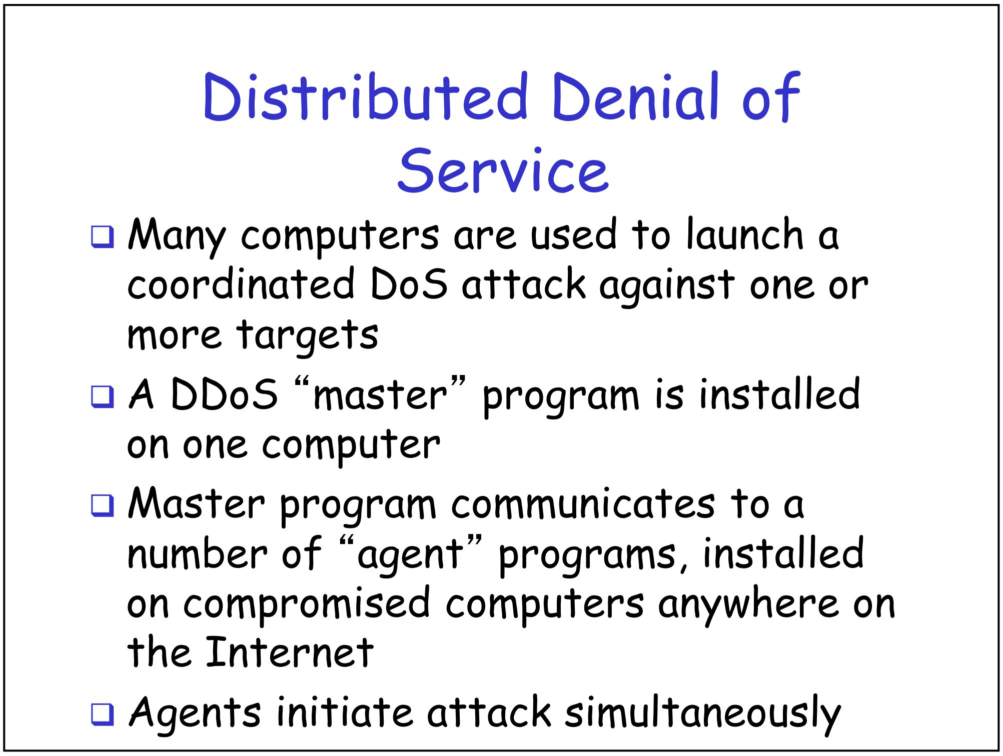
So to run my attack I am just going to get a lot a help (from people who might not realise they are helping me)

Give example Alice and Bob, where Alice phones 100 take-away restaurants places to deliver food to Bob at 9pm on one particular evening.

The master: Alice,

the agents/slaves: pizza place

Victim: bob



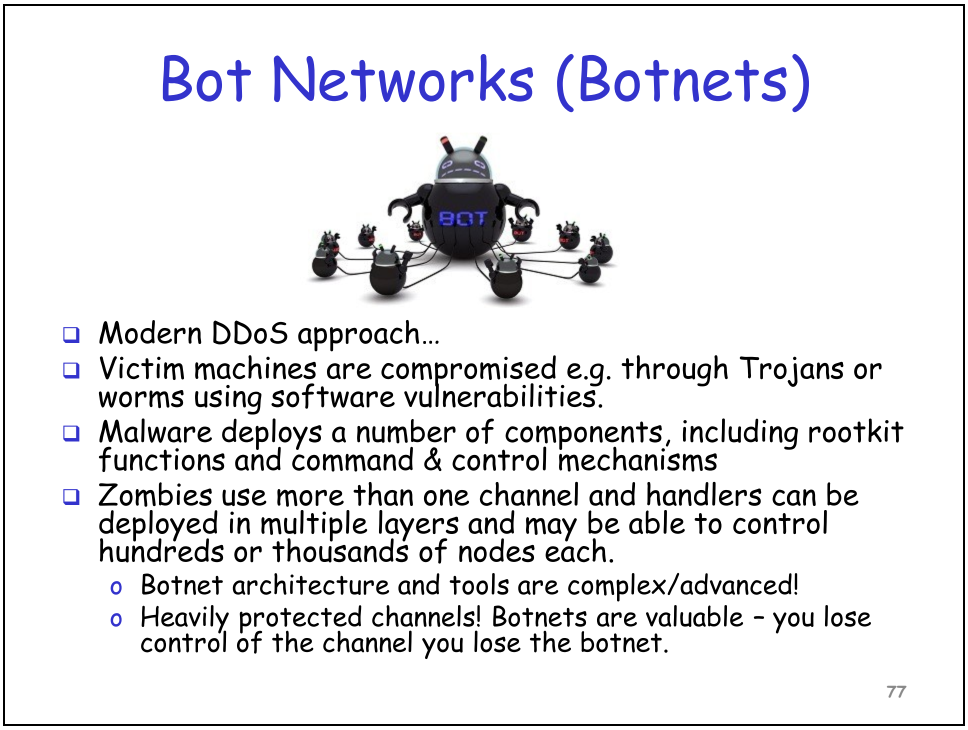
Distributed Denial of Service

- ❑ Many computers are used to launch a coordinated DoS attack against one or more targets
- ❑ A DDoS “master” program is installed on one computer
- ❑ Master program communicates to a number of “agent” programs, installed on compromised computers anywhere on the Internet
- ❑ Agents initiate attack simultaneously

DoS attacks launched from a single computer

Hundreds or thousends of agents

Break into hundreds or thousands of machines all over the Internet



Bot Networks (Botnets)

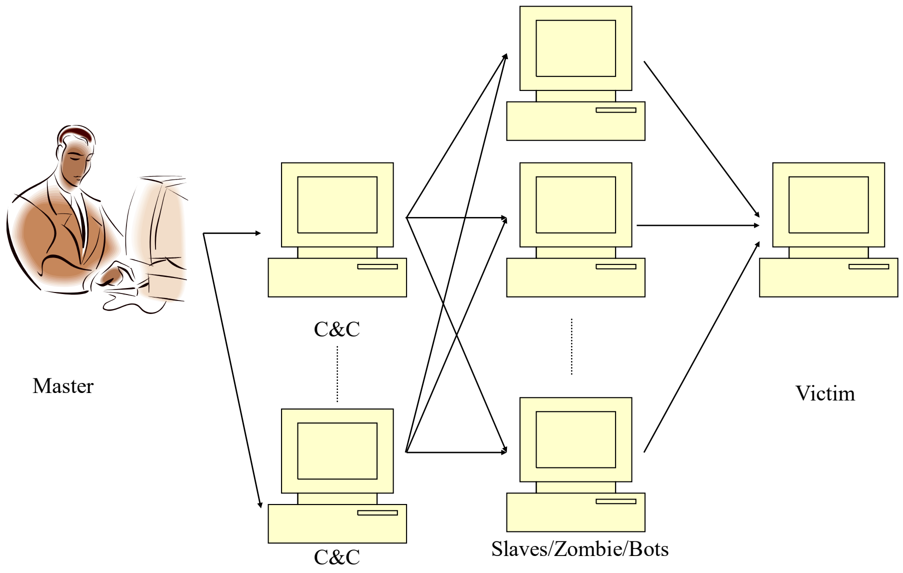
- Modern DDoS approach…
- Victim machines are compromised e.g. through Trojans or worms using software vulnerabilities.
- Malware deploys a number of components, including rootkit functions and command & control mechanisms
- Zombies use more than one channel and handlers can be deployed in multiple layers and may be able to control hundreds or thousands of nodes each.
  - Botnet architecture and tools are complex/advanced!
  - Heavily protected channels! Botnets are valuable – you lose control of the channel you lose the botnet.

Look for issue, compromise – automated process.

What does the root kit do?

A rootkit hides the existence of other processes or programs.

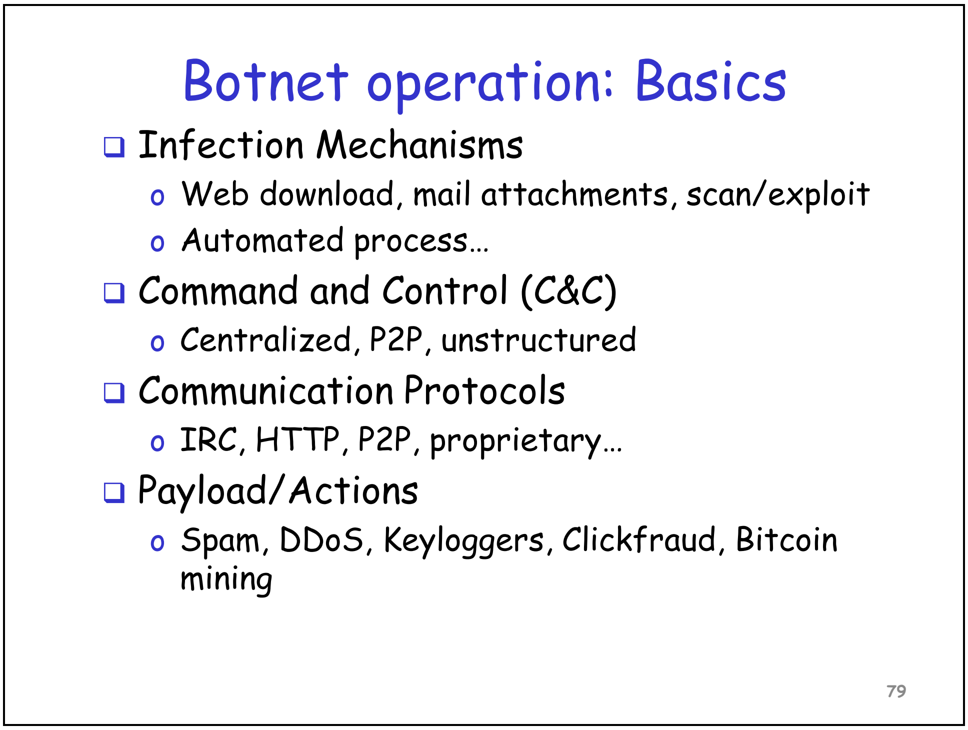
Botnet



Hundreds or thousands of slaves

Often with IP spoofing

This is a flat architecture but in reality this can be hierarchical in terms of control, P2P/mesh etc



Botnet operation: Basics

- Infection Mechanisms
  - Web download, mail attachments, scan/exploit
  - Automated process…
- Command and Control (C&C)
  - Centralized, P2P, unstructured
- Communication Protocols
  - IRC, HTTP, P2P, proprietary…
- Payload/Actions
  - Spam, DDoS, Keyloggers, Clickfraud, Bitcoin mining

Infection

Web-based infection – user downloads malware (Drive by download -> unintended download)

Mail attachments – in itself a product from spam being received, phishing

Scan, exploit and compromise – botnet looks for vulnerable hosts, exploits this vulnerability to compromise host.

Once a machine is infected the botmaster can issue it instructions.

Command and Control

Centralized - A centralized model is characterized by a central point that forwards messages between clients. The centralized model has some advantages such as simple implementation and customization. However, the centralized C&C model will be detected and destroyed easier (traffic analysis will identify this single target for messages).

P2P – botmaster commands can be issued from any peer, any node can be controller. More complex to design but more robust to analysis.

Unstructured – bots do not actively talk to the botmaster, only listen for incoming messages. Botmaster randomly scans for bots, and sends command if it finds one. You have very high data latency, but high survivability.

Communication Protocols

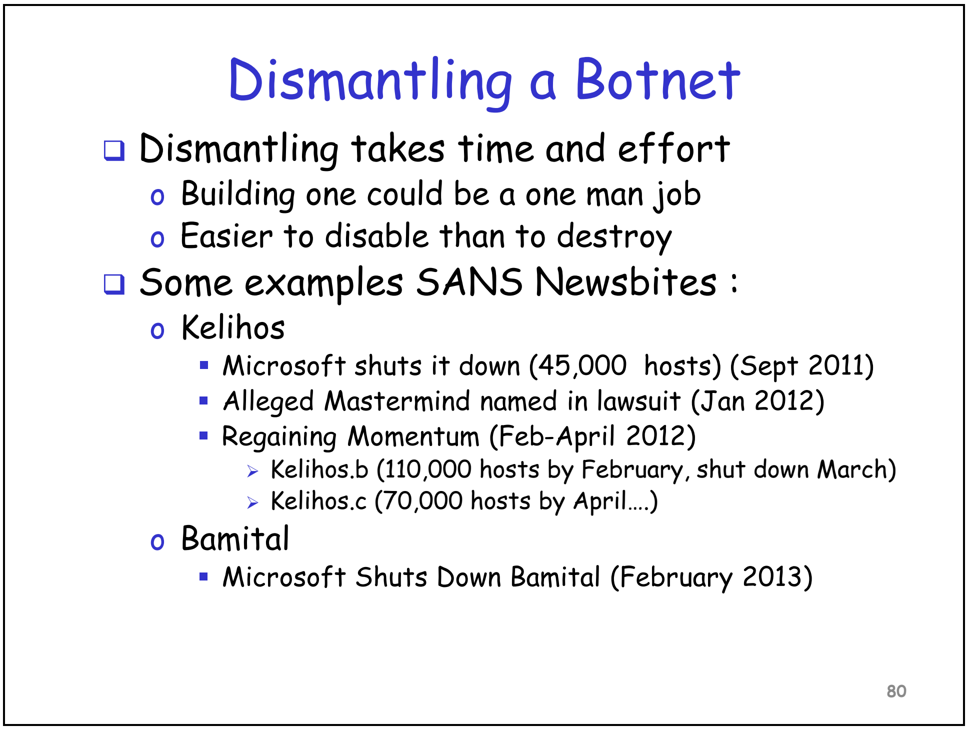
Important here is that bots can use existing protocols

IRC – good for point to multipoint and point to point communication

HTTP – generally the bot will periodically check a web server for commands

P2P – proprietary communication.

Ways that different bots are designed feature in what they are useful for. You would think that a spam bot would have some way of getting commands from the botmaster (but there is not much it wishes to send the botmaster in return, this might be ideal for unstructured control), similarly a keylogger really needs to send the botmaster data (but once it is on the infected machine the botmaster does not really need to tell it what to do).



Dismantling a Botnet

- ❑ Dismantling takes time and effort
  - o Building one could be a one man job
  - o Easier to disable than to destroy
- ❑ Some examples SANS Newsbites :
  - o Kelihos
    - Microsoft shuts it down (45,000 hosts) (Sept 2011)
    - Alleged Mastermind named in lawsuit (Jan 2012)
    - Regaining Momentum (Feb-April 2012)
      - Kelihos.b (110,000 hosts by February, shut down March)
      - Kelihos.c (70,000 hosts by April….)
  - o Bamital
    - Microsoft Shuts Down Bamital (February 2013)

Dismantling is a hard task, requiring much resources. Building one on the other hand could be simple.

The problem with botnets are that even when they are detected, and you find a way of shutting them down (basically make it stop sending spam, take over control and tell bots to send nothing or send to a sink)…the malware is not necessarily removed from the host. That is not practical, these can be all over the place? So really they are only dormant.

Lets look at examples

First an example of how difficult it is

--Microsoft Shuts Down Kelihos Botnet

(September 27, 2011)

Microsoft and Kaspersky stop botnet activity - have some domains shut down, get the infected machines to talk to a server they controlled.

--Microsoft Names Alleged Kelihos Botnet Mastermind in Lawsuit
(January 24 & 25, 2012) Microsoft has filed a lawsuit in US District Court in Alexandria, Virginia, naming the individual it believes was responsible for operating the Kelihos botnet. So all this effort by two quite large companies to take down a botnet built by a single person.

--Kelihos Botnet Regaining Momentum

(February 1 & 3, 2012)

The Kelihos botnet appears to be regaining its foothold. The malware was never removed from the machines, and although it was possible through botnet control commands this operation would have been illegal in some countries where infected machines were located. In this case user notification equally difficult – know the host not the owner.

Now for a happier story

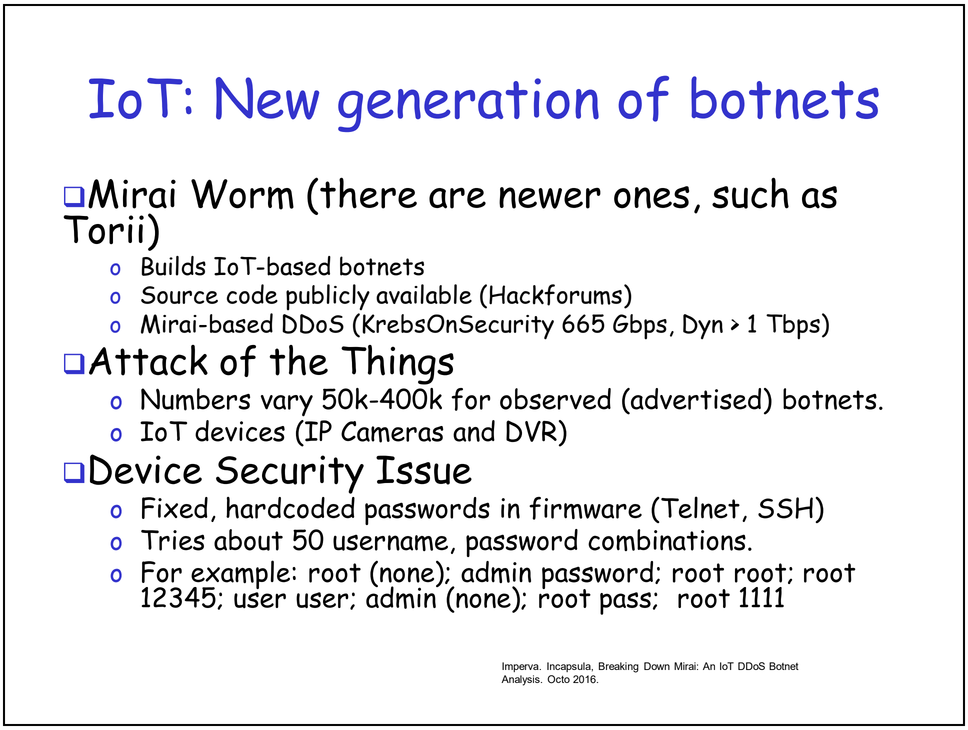
Microsoft and Symantec take down Bamital.

Bamital was a click fraud botnet – it was used to hijack web searches from legitimate search engines and victim were redirected to fake listings with the goal of getting the victim to click on adds.

In this case take down might be more permanent.

1.physical take down of infrastructure US federal marshals.

2.Notification to victims that their system is compromised – instead of being redirected to a bogus search listing the victim is now directed to a microsoft site stating their machine in infected and which gives instructions to fix the problem.



IoT: New generation of botnets

- ❑Mirai Worm (there are newer ones, such as Torii)
  - o Builds IoT-based botnets
  - o Source code publicly available (Hackforums)
  - o Mirai-based DDoS (KrebsOnSecurity 665 Gbps, Dyn > 1 Tbps)
- ❑Attack of the Things
  - o Numbers vary 50k-400k for observed (advertised) botnets.
  - o IoT devices (IP Cameras and DVR)
- ❑Device Security Issue
  - o Fixed, hardcoded passwords in firmware (Telnet, SSH)
  - o Tries about 50 username, password combinations.
  - o For example: root (none); admin password; root root; root 12345; user user; admin (none); root pass; root 1111

Imperva. Incapsula, Breaking Down Mirai: An IoT DDoS Botnet Analysis. Octo 2016.

Torii is latest IoT malware

- Lot of payloads (more flexible botnet)

- Support different host architecture (any modern computer, smartphone, and tablet)

-Very persistent and stealthy

-Still used weak credentials to spread.

More info

https://www.sentryo.net/the-mirai-iot-botnet-a-publically-available-turn-key-threat-2/

http://www.cbc.ca/news/business/several-baby-monitors-vulnerable-to-hacking-cybersecurity-firm-warns-1.3213046

https://www.csoonline.com/article/3310222/security/new-vicious-torii-iot-botnet-discovered.html

The end!

?

Any questions…