Networks and Communications

“Network Security”

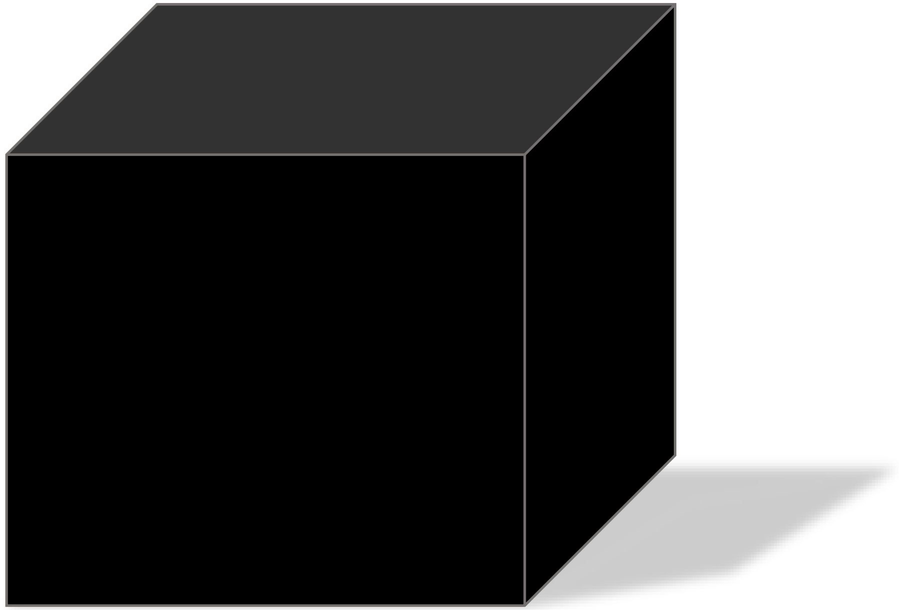
Konstantinos Gkoutzis
Imperial College

Outline

- Terminology
- History
- Encryption
- Key Distribution
- Access Control
- + more

The Internet as we know it

- The Web
- Video/Audio Streaming
- Online Games
- Cloud Apps
- [...]



The Internet as we don’t (?) know it

A Scary New Kind of Malware Is Invading Banks All Over the World

New research from Kaspersky claims that over 140 institutions—including banks, government organizations, and telecom companies—have been infected with invisible malware that hackers are using to suck money out of bank accounts.

Malware that skulks in memory, invisibly collecting sysadmins' passwords

Banks, telecommunication companies and government organisations in the US, South America, Europe and Africa have already been hit by the ongoing (and stealthy) attacks.

Wednesday 08 February 2017

Sloppy iOS apps expose 'encrypted' user traffic

76 iOS applications with an accumulated 18 million downloads between them are vulnerable to having their encrypted HTTPS traffic compromised due to bad TLS cert handling which escaped Apple's attention

Macro Malware Comes to macOS

Macro-based malware has crossed the divide between the Windows and Mac platforms. A cybercrime group whose command and control infrastructure resolves to an IP address geo-located in Russia is using a Word document laced with a malicious macro that executes solely on macOS.

David Beckham Hires Cybersecurity Expert To Probe Email Leak

18.6 million emails were stolen and leaked from PR firm, including allegedly doctored messages made to damage Beckham's reputation.

- **These were just yesterday..!**
- Who does all these things?

Russia Detains Nine 'Hackers' Over $17 Million Bank Thefts

Russia has detained nine people alleged to be part of a cybercrime ring accused of stealing some $17 million dollars from bank accounts, the interior ministry said Wednesday.

Dutch Voter Guide Website Leak Highlights Privacy Concerns

A data leak from StemWijzer, a Dutch voter guide website, has raised questions about its intentions and whether it is quietly conducting popularity polls and infringing upon voters' privacy, Reuters reports.

H/P/V/A/C

- Popular online term(s) until the early 90s
  - *then the media just started calling everyone a "hacker"*
- **H:** Hackers
- **P**: Phreakers
- **V**: Virii (creators)
- **A**: Anarchists
- **C**: Crackers
- *See also: "The Hacker Crackdown"*

Hackers

- **H:** Hackers

- Originally meant: highly competent (computer) engineers who explore different ways of using/combining things

- Since then: "*dangerous criminals who are after your data!!1*"
  - a.k.a. "*cybercriminals*", or "*cyberterrorists*"

- Luckily, we still use this term with its old meaning
  - e.g. IC Hack

- Can be divided into: White/Gray/Black Hats

- Example: Samy Kamkar

Phreakers

- **P:** Phreakers
- Originally meant: **Ph**one+Hac**kers**
- Since then: the telephone network is now almost fully digital (*and online*)
  - so now they are just called hackers
- Example: John Draper (*a.k.a. Captain Crunch*)

Virii (creators)

- **V:** Virii (creators)

- Creators of computers viruses

- Why?
  a) Because they are curious
  b) Because they can
  c) Because £☒¥€$

- Popular modern viruses:
  - Ransomware (*e.g. using Bitcoin*)
  - Spyware (*e.g. browser add-ons*)
  - Trojans (*e.g. remotely controlled botnet zombie*)

- Example: David L. Smith

Anarchists

- **A:** Anarchists
- Originally meant: physical security perpetrators who organise their attacks online
  - e.g. BBS/Forums/IRC
- Since then: when they are peaceful, they are usually called "*hacktivists*"
- Example: Some parts of Anonymous

Crackers

- **C:** Crackers

- Originally meant: wannabe hackers, who use the tools of others to infiltrate systems

- Since then: mostly confused with "*hackers*"
  - sometimes confused with "*code-crackers*"

- More crackers than hackers in the world
  - (*mostly up to no good*)

- Can sometimes hire Black Hat hackers to deliver "hacks"

- Example: Most modern organised crime groups

Others

- Warez scene
  - anything that is uploaded or downloaded illegally
  - today we call them “pirates”
  - Example: most of the files on PirateBay
- Social Engineers / Phishers / Catfishes
  - manipulators; attack “personal/human security”
  - pretend to be someone else; try to confuse/scam you
  - Example: Kevin Mitnick
- DDoSers
  - a Distributed Denial of Service attack participant
  - usually using the tools of others
  - Example: Low Orbit Ion Cannon

Others (cont'd)

- Whistleblowers
  - former "insiders" of companies/organisations
  - reveal secrets, even after signing NDAs (*Non-Disclosure Agreements*)
  - Example: Edward Snowden

- Spammers / Botters
  - mass-senders of unsolicited (spam) messages
  - these days, they mostly use botnets/zombies
  - Example: (*list on SpamHaus*)

- Cyberbullies
  - *a.k.a. "trolls"*
  - aim to harass, stalk, offend, and even threaten online users
  - Example: most YouTube commenters
  - If this happens to you, **please report it to us**

On Laws

Imperial College London

- Alas, are we helpless?
- Computer Misuse Act (1990)
- Copyright, Designs and Patents Act (1988)
- Criminal Justice Act (2003)
- Data Protection Act (1998)
- Defamation Act (2013)
- Disability Discrimination Act (1995)
- Digital Economy Act (2010)

On Laws (cont'd)

- e-Commerce Regulations Directive (2002)
- Freedom of Information Act (2000)
- Obscene Publications Act (1959)
- Protection of Children Act (1978)
- Regulation of Investigation Powers Act (2000)
- + EU Cybercrime laws
  - *(for a little while longer)*
- + any laws of the country/ies where involved hosts are physically located
  - e.g. DMCA

On Standards

- IANA
  - Internet Assigned Numbers Authority
- ICANN
  - Internet Corporation for Assigned Names and Numbers
- IETF
  - Internet Engineering Task Force
- ISOC
  - Internet Society
- EFF
  - Electronic Frontier Foundation
- W3C
  - World Wide Web Consortium
- ISO
  - International Organization for Standardization

What else is there?

- Since laws and standards can be ignored by malicious users...
- ...we have created ways of protecting our data and our systems from attackers
- **Digital Attacks**:
  - Accounts, firewalls, antivirus applications, cryptography, backup, *+more*
- **Physical Attacks**:
  - Locks/keys, cameras, sensors, biometrics scanners, alarms, guards, *+more*
- **Personal/Human Attacks**:
  - DBS checks, staff training, Internet traffic logging/monitoring, *+more*

You may want to follow:

- Schneier on Security
- Naked Security by Sophos
- ThreatPost and SecureList by Kaspersky
- TheRegister (Security section)
- Wired (Security section)
- Hacker News
- CSO Online
- **Digital Attack Maps** by NorseCorp and ArborNetworks

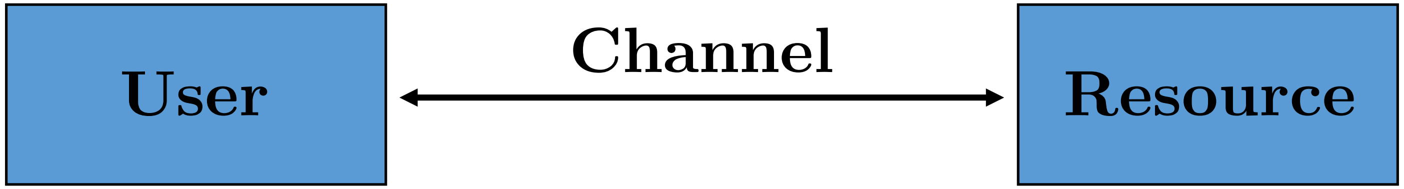
Can you break this?

- Someone sent you this encrypted message:

FRPHEVGL

- What does it mean?
  - How was it encrypted?

Network Security Issues

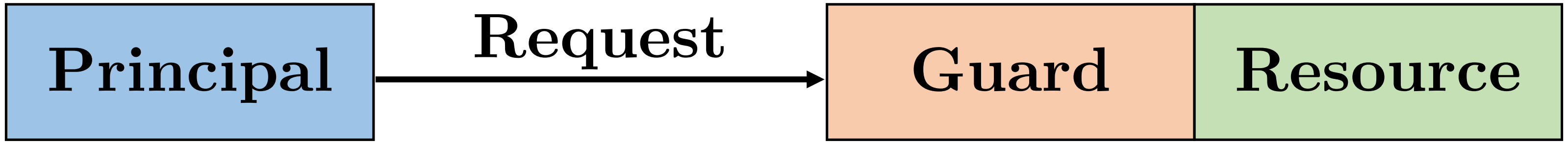


- *Access Control*
  - Only *certain* users are allowed access to a resource
- *Authentication*
  - User knows that the (re)source really is what it says it is, and vice-versa
- *Confidentiality*
  - Users limit access to information/resources they own
  - Data confidentiality; Traffic confidentiality
- *Integrity*
  - Actions of a user should not be able to affect the overall integrity of a resource
- *Non-Repudiation*
  - Users cannot deny communication took place (*really "in fashion" at the moment*)

Security Aspects

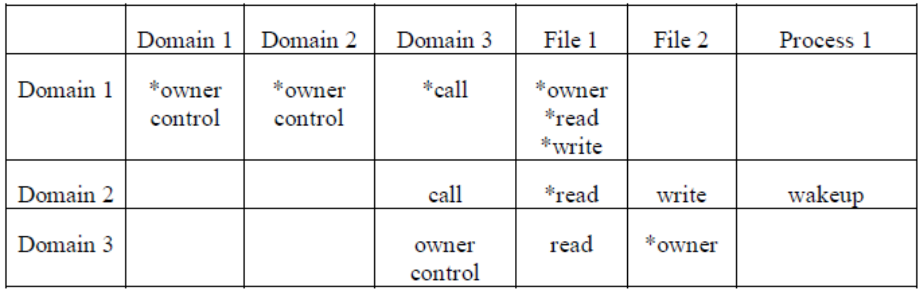
- To deal with security we need:
  - **Access Control**
  - **Security Policy**
  - Technical infrastructure for implementing said *Policy*, using:
    - **Secure Channels**, where
      - Users and their data are *authenticated*
      - Information they exchange is *confidential*
  - **Monitoring / Logging / Auditing**

Access Control



- Assuming a *secure* channel, the *Guard* controls:
  - Which Principals can access the Resource
  - Where Principals are allowed to be located
  - What Requests Principals are allowed to make

- *See Lampson's Protection for the "Access Matrix"*

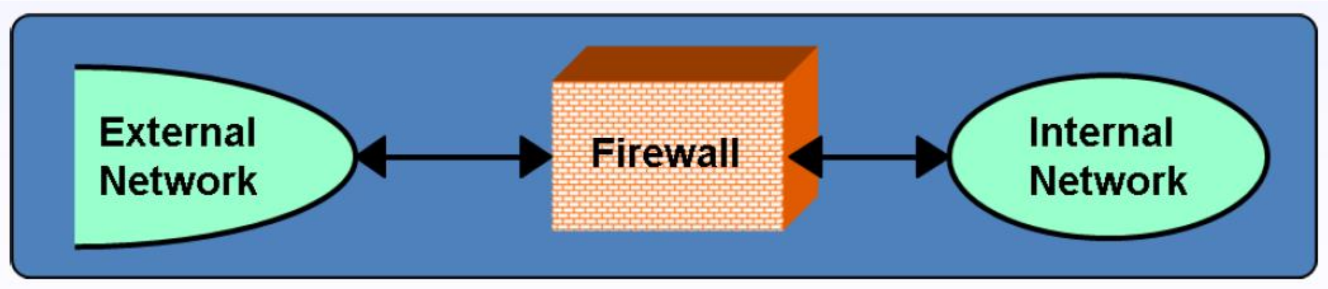


	Domain 1	Domain 2	Domain 3	File 1	File 2	Process 1
Domain 1	*owner control	*owner control	*call	*owner *read *write		
Domain 2			call	*read	write	wakeup
Domain 3			owner control	read	*owner	

Firewalls

- Ensuring that all hosts are secure is a complex process
  - Heterogeneous systems => different configurations
  - Users can be careless
  - Even host managers/administrators can be careless

- ***Firewalls*** control access to the network => a security gateway between the *internal* and *external* networks
  - Application level Gateway (*e.g. netfilter's iptables, SpyShelter/Comodo*)
    - Proxies (*e.g. SOCKS, HTTP*)
  - Circuit Level Gateway (e.g. Tor)
  - Packet Filtering
    - Stateful Multilayer Inspection
  - Hybrid (*combination of the above*)



- Can be purely *software-based*, or even *hardware-based*
  - can replace/be a router between public and private networks

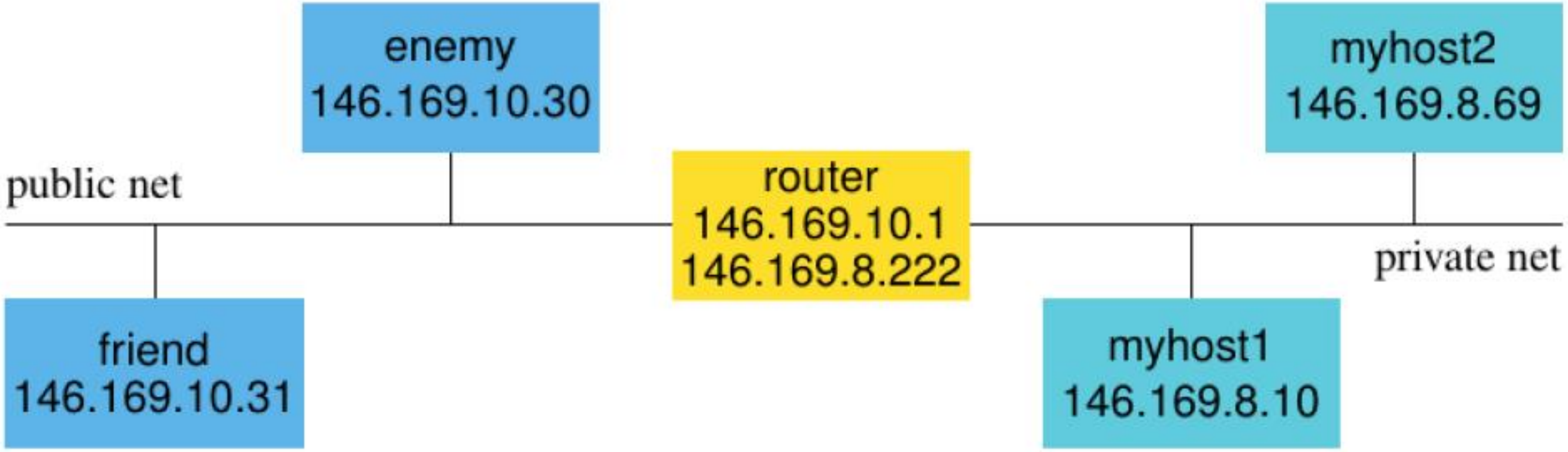
Firewall Components

- Task: to analyse inbound packets and, based upon existing rules, decide whether to block or allow each packet
- **Application-level gateway**
  - runs on the host; only protects that host
- **Proxy server**
  - runs on the network; can protect entire LAN
- **Circuit-level gateway**
  - acts like a (*non-caching*) proxy, viz. it fully takes over the host's communication with the recipient, and then decides what to allow/block
- **[Stateful] Packet Filtering**
  - stateless: checks source/destination IP addresses and source/destination ports
  - stateful: remembers connections and checks contents of current *and* previous packets

Open Internet Access

Imperial College London

- No firewall:



- Anyone who knows your public IP can contact you
  - (*and even if they don't, they can randomly end up on it*)

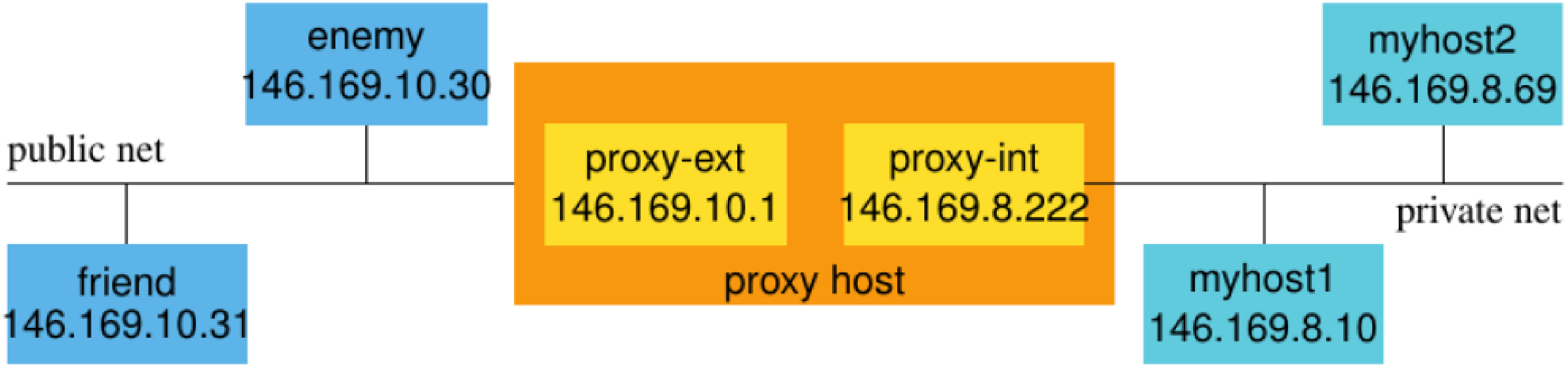
Access via Proxy

Imperial College London

- Proxy can filter incoming/outgoing traffic

- Different modes:
  - **Normal**: the client is aware (*and needs to be set up*)
  - **Transparent**: the client is unaware (*the local router takes care of everything*)
  - **Reverse**: runs on the receiving side, "impersonating" servers (*CDN load balancing*)

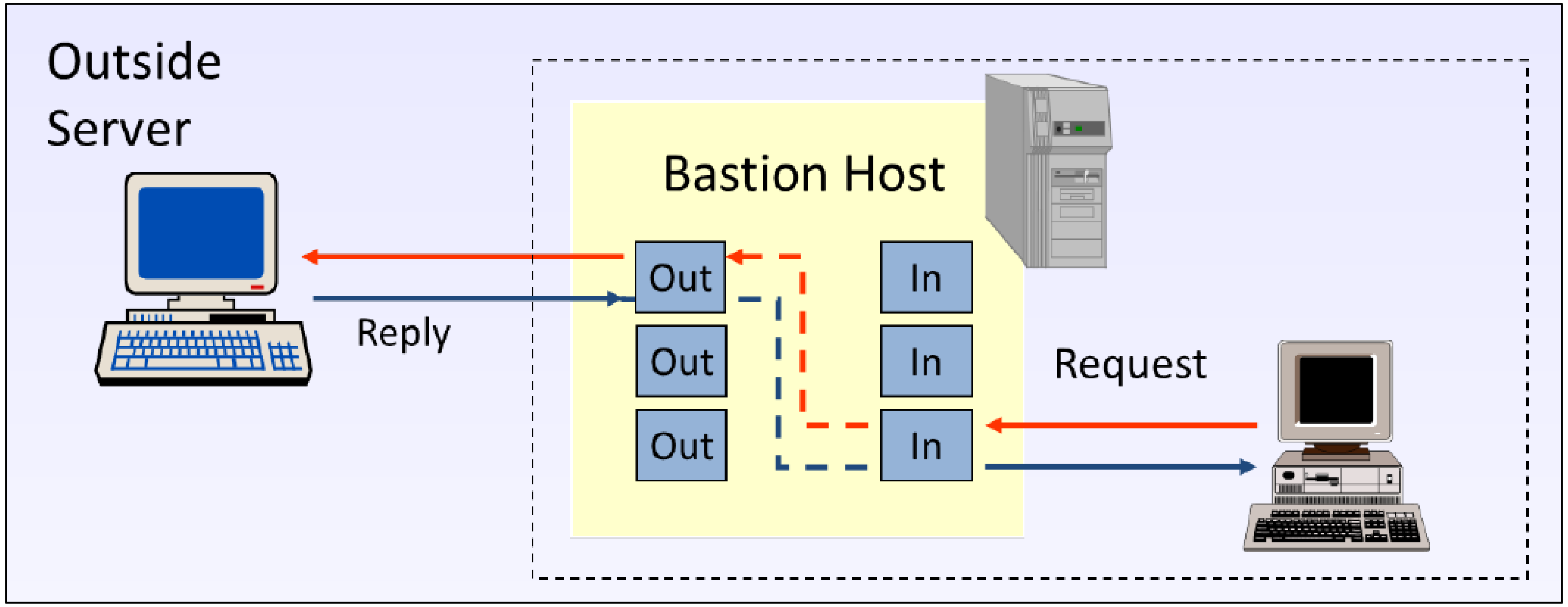
- Private network only accessible via proxy:



Bastion Host

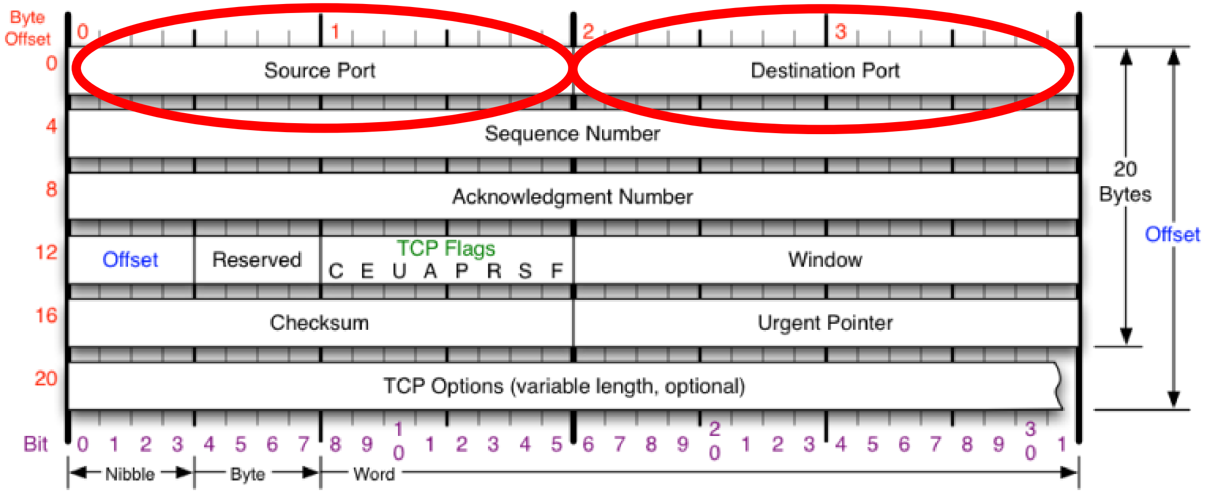
- Expects to be attacked..!
- Performs auditing/logging
- Should run a trusted/secure OS
- Administered via a dedicated terminal
- Only runs necessary software/services – minimal OS
  - Remove non-essential applications, utilities, services (*e.g. X11*)
  - Set file permissions, turn on file quotas, process limits, etc.
  - No regular user accounts
  - No NFS mounts
  - Make filesystem(s) read-only, if possible

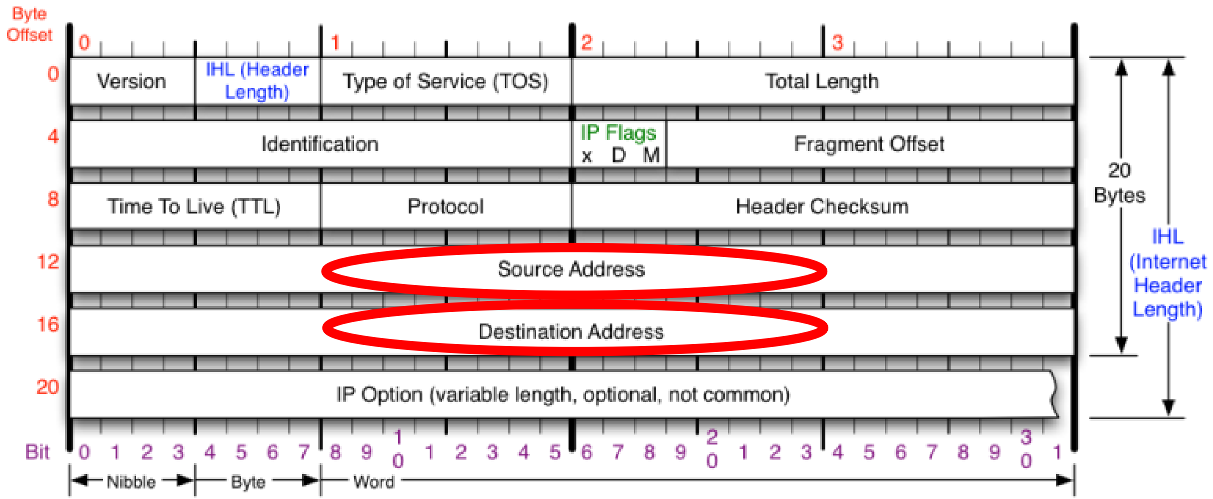
Stateful inspection firewalls

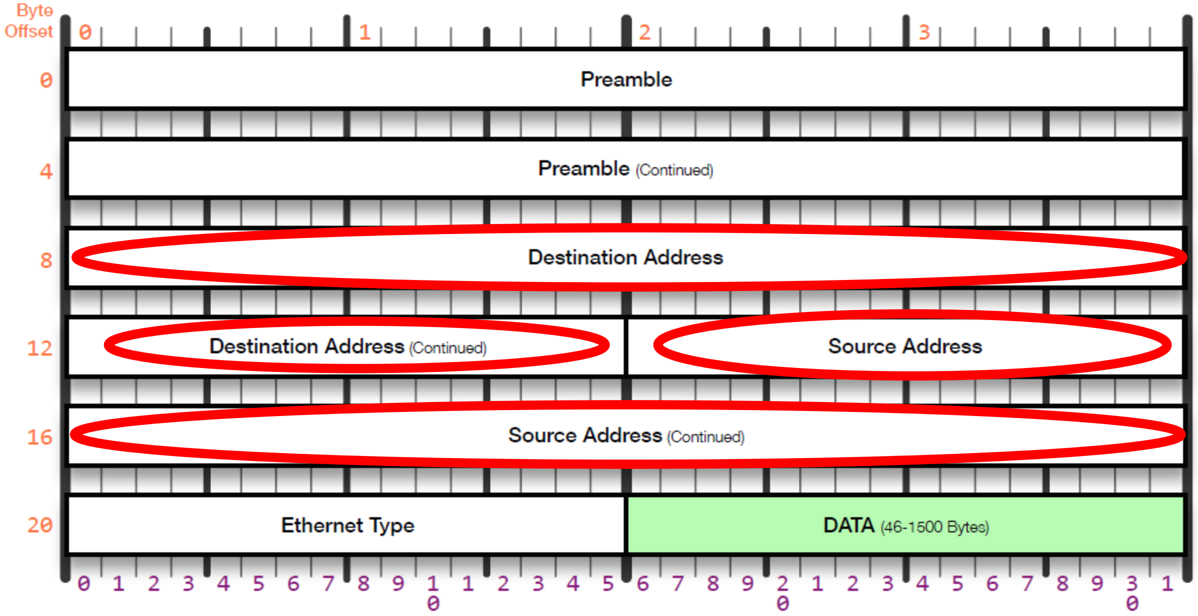


- Relays connections and maintains connection state
- Can also authenticate users
- Can drop connections based on destination, incorrect connection packets, time, volume, etc.
- Useful for logging/auditing/monitoring

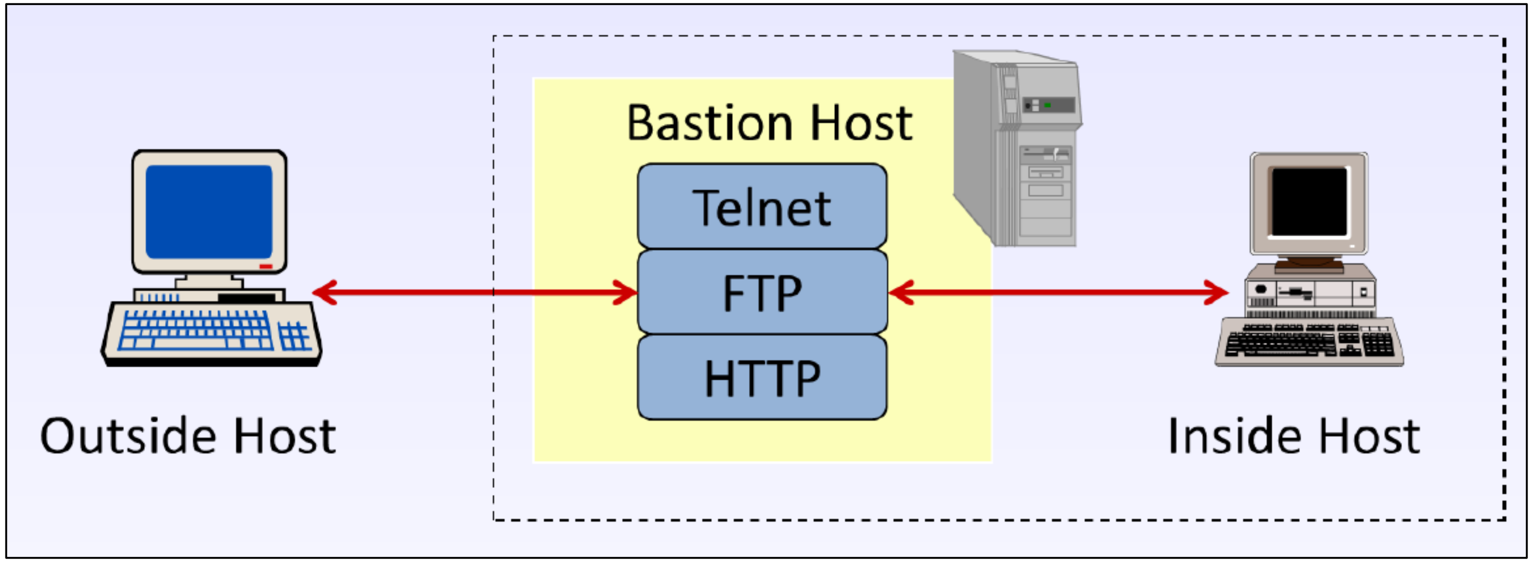
Packet Filtering firewalls







Application-level Gateways

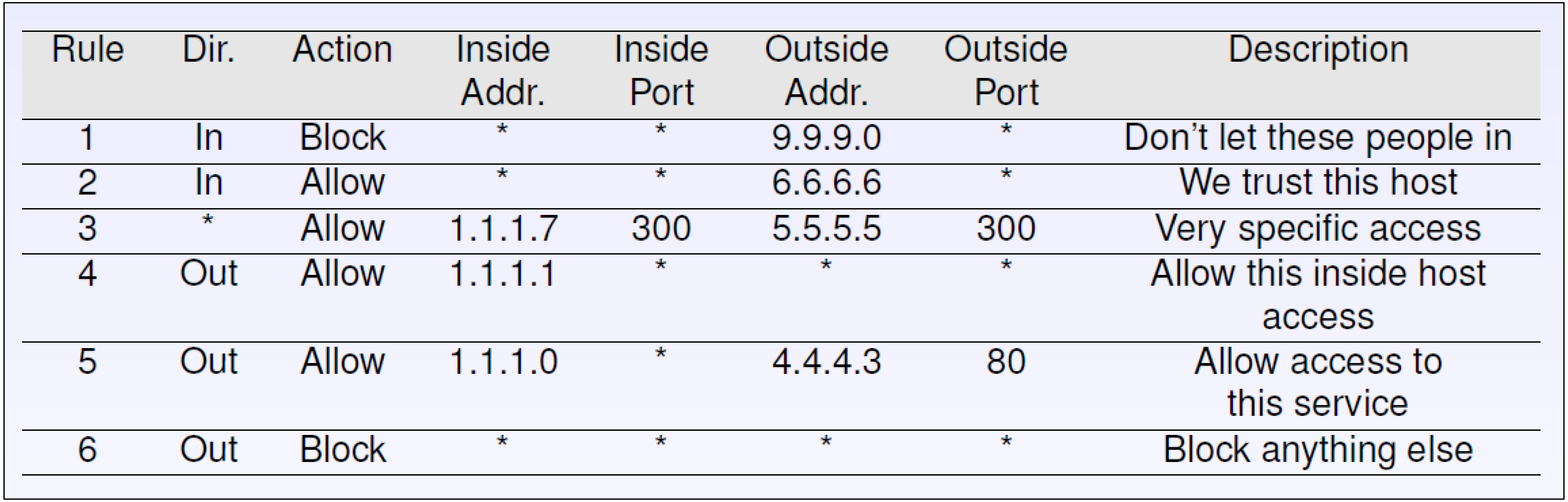


- a.k.a *Proxy firewalls*
- In the midst of a “logical” connection, thus allowing it to monitor traffic
- Can block/filter/report based on app-level msg. content
- Can scan for data leaks, worms/viruses, etc.
- Can rewrite data (!)

Access Control Lists

Imperial College London

- Packet Filtering Rules:



Rule	Dir.	Action	Inside Addr.	Inside Port	Outside Addr.	Outside Port	Description
1	In	Block	*	*	9.9.9.0	*	Don’t let these people in
2	In	Allow	*	*	6.6.6.6	*	We trust this host
3	*	Allow	1.1.1.7	300	5.5.5.5	300	Very specific access
4	Out	Allow	1.1.1.1	*	*	*	Allow this inside host access
5	Out	Allow	1.1.1.0	*	4.4.4.3	80	Allow access to this service
6	Out	Block	*	*	*	*	Block anything else

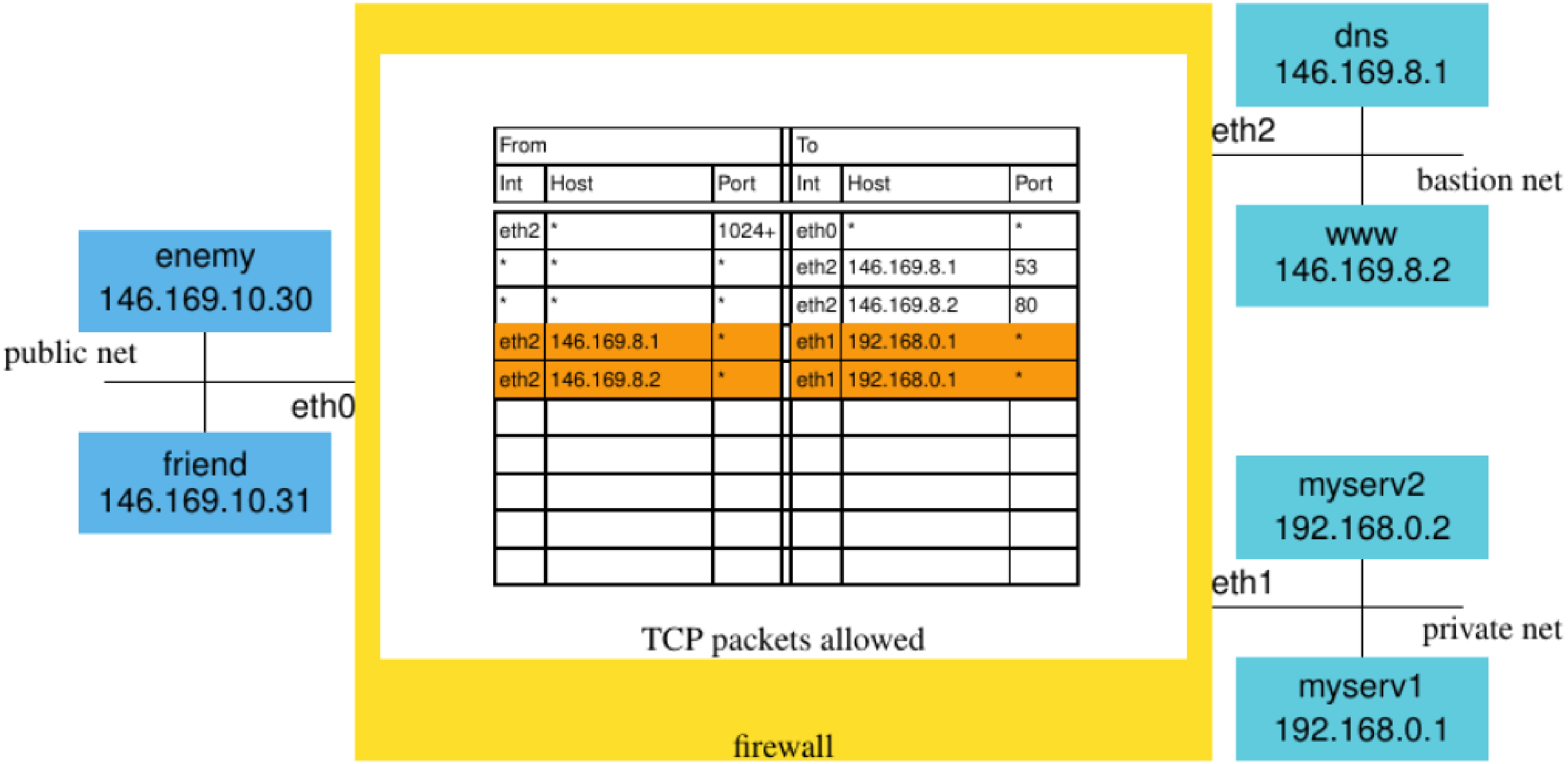
- Rules are checked from top to bottom until a match is found

Firewall Example (with ACL)

- All non-well-known server ports on the bastion network can access all hosts on the public net

- The DNS server running on host “dns” and HTTP server running on machine “www” can be accessed from all other networks

- Host “dns” and “www” may contact any port on “myserv1”

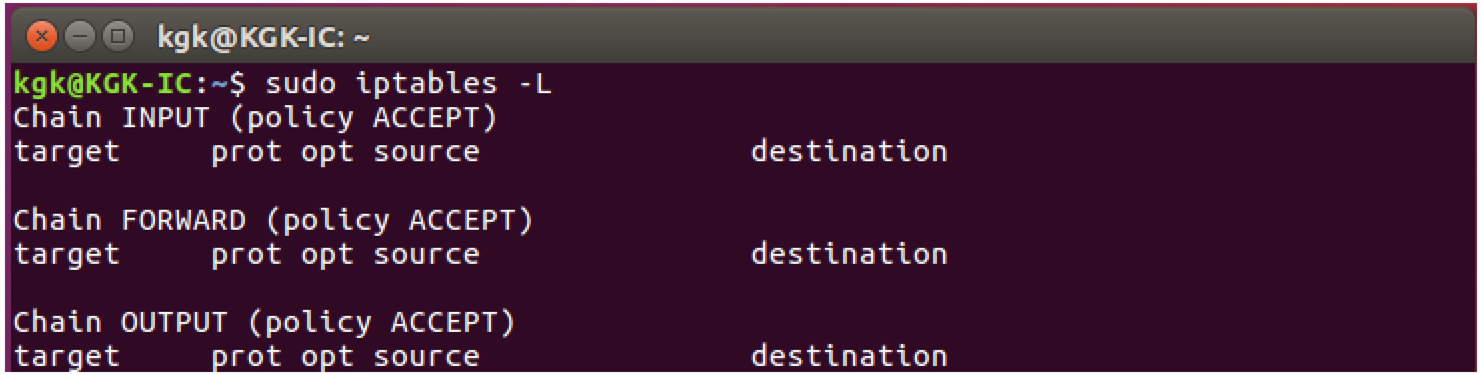


From			To		
Int	Host	Port	Int	Host	Port
eth2	*	1024+	eth0	*	*
*	*	*	eth2	146.169.8.1	53
*	*	*	eth2	146.169.8.2	80
eth2	146.169.8.1	*	eth1	192.168.0.1	*
eth2	146.169.8.2	*	eth1	192.168.0.1	*

TCP packets allowed

Firewall Example (cont'd)

- **iptables**
  - Administrative tool for packet filtering
  - Consists of *chains* with *filter rules* in *tables*



```
kgk@KGK-IC:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

- **tcpd**
  - Daemon controlling access to Unix services
  - Consults two files: `/etc/hosts.allow` and `/etc/hosts.deny`

IDS, IPS, NGFW, UMT

- **IDS**: Intrusion Detection System
  - software that detects intrusions (*e.g. identifies a DDoS attack*)
  - but does nothing to stop them
  - except inform the system

- **IPS**: Intrusion Prevention System
  - software that prevents intrusions (*e.g. actively blocks SYN flooders*)
  - either includes, or works with, an IDS

- **NGFW**: Next Generation Firewall
  - a (statefull) firewall that came with an IPS/IDS system
  - (*in addition to ACL mechanisms*)

- **UMT**: Unified Threat Management
  - similar to NGFW, but with more capabilities (*e.g. antispam/antivirus*)

DMZ

- DeMilitarized Zone
- The “area” between you and the outside (*dangerous*) world
  - the “neutral zone”
- External hosts can only speak directly to your internal hosts that lie within the DMZ (*if any*)
- All other, non-DMZ, hosts are hidden/protected by the gateway/router/firewall
- The router uses NAT (Network Address Translation) to get the external messages to the correct internal host
- If you want to expose an internal host without putting it in the DMZ, you can use “port forwarding”

Port Forwarding

- This lets the router know that packets for certain ports should be forwarded directly to an internal host/port

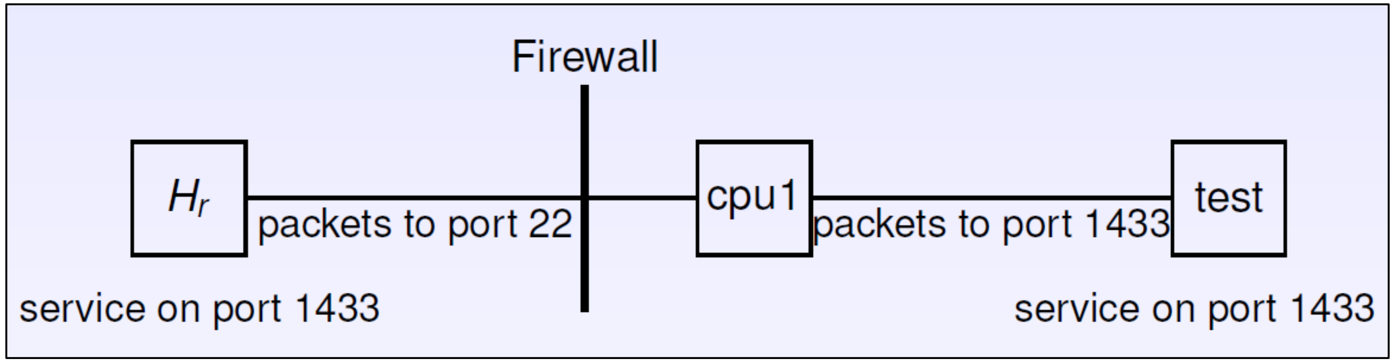
- e.g. any packet received at port 12345 of the router's public IP, should be forwarded to the NAT-based LAN IP of host1 at its port 80

- So if I access `http://yourpublicip:12345/` I am really accessing `http://host1:80`
  - but I was not able to do this directly since host1 is hiding behind the router
    - and thus does not have a public IP

- Very useful if you wish to host servers (*which you want to access on the Internet*)
  - or if some application is requesting non-standard ports (*e.g. games*)

Getting Around Firewalls

- **ssh**



- Often, most non-standard services are blocked by a firewall

- If ssh is allowed, you can use it to tunnel through a firewall

- H_r executes `ssh -g -N -L 1433:test:1433 user@cpu1`
  - H_r connects to cpu1 on port 22
  - cpu1 connects to test on port 1433
  - ssh on H_r provides a service on port 1433 (*via 22*)

- (*Same applies for Remote Desktop or VNC*)

Getting Around Firewalls (cont'd)

- One could potentially "spoof" a MAC address
  - you can easily "rewrite" your MAC on your software
- One could also attempt to "spoof" an IP address
  - but a stateful firewall will *probably* catch it
- You could also use a VPN (Virtual Private Network) to "tunnel" around a firewall
  - the firewall won't be able to know what you are doing
    - as long as your *tunnel* is secure (*e.g. using SSL – Secure Sockets Layer*)
- However, firewalls can learn to block your secured VPN connections
  - exactly because they cannot read them..!
- It may also be against the Acceptable Use Policy of the network

Security Policy

- Each company/organisation needs to define its Network/IT Security Policy

- Useful standard: ISO/IEC 17799:2005
  - *IT – Security Techniques – Code of Practice for Information Security Management*

- e.g.
  - Intranet/Internet access rights
  - Allowed use of software/hardware
  - Risk assessments
  - Training
  - +more

- Usually the role of the IT/Network Manager/Director

“The only secure computer is one that's unplugged, locked in a safe, and buried 20 feet under the ground in a secret location... and I'm not even too sure about that one.”

(attributed to Dennis Hughes – the first Chief of the Computer Investigations Unit at the FBI)

Cryptography

- Cryptography
- Κρυπτογραφία

- κρυπτο- = hidden

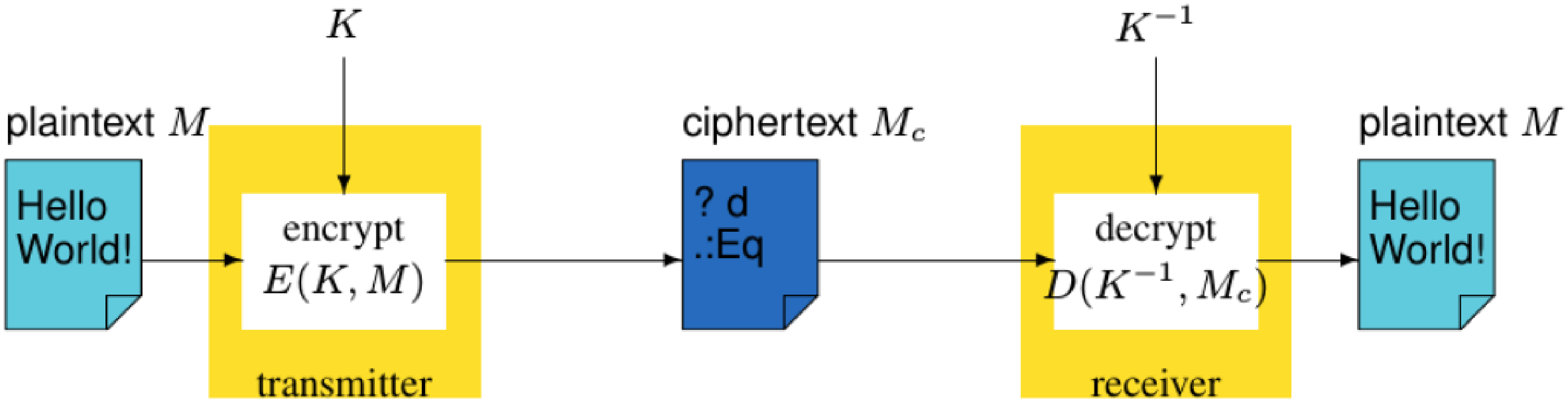
- -γραφία = writing

- Writing something “in code” (=to encode *or* to encrypt)
  - using a specific algorithm (=series of steps)
    - while retaining the ability to retrieve it afterwards (=to decode *or* to decrypt)

- e.g. **SECURITY** with the *Caesar Cipher* and a parameter of **-13** becomes **FRPHEVGL**
  - i.e. we shifted all letters up by 13 places
    - this is a simple “shift” algorithm, but there are a lot more complex ones
      - *cipher = cryptographic algorithm*

Secure Channels: Encryption

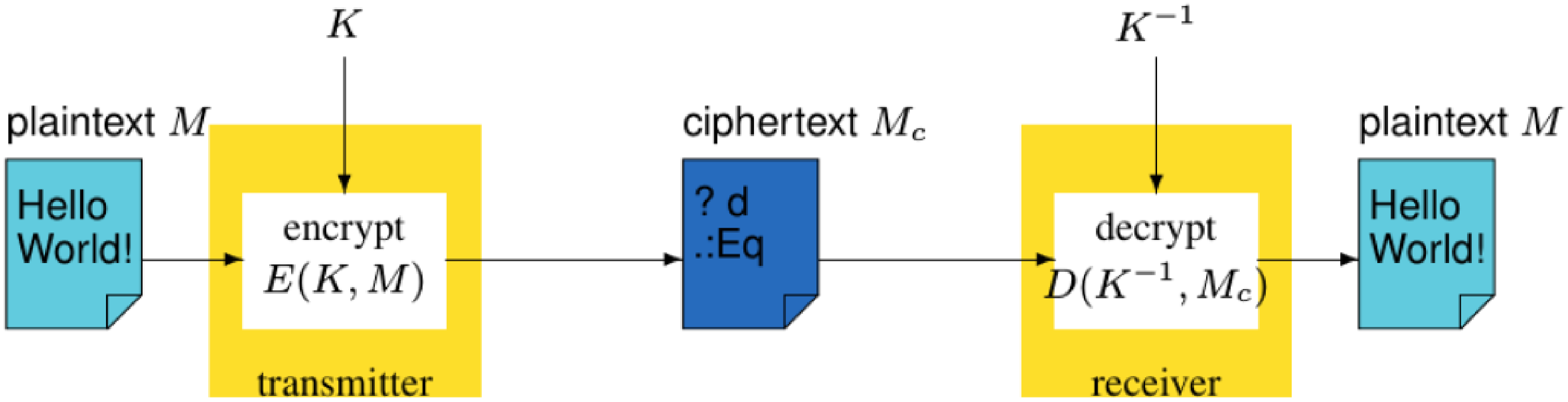
- Assumptions
  - Physical channel is open to attack from the enemy
  - Enemy may read and/or alter any bit pattern



- Ciphertext $\mathbf{M_c} = \mathrm{E(K,M)}$
- Plaintext $\mathbf{M} = \mathrm{D(K^{-1},M_c)}$
- Only certain users should have K and K^{-1}

M: Message
K: Key
E: Encrypt
D: Decrypt

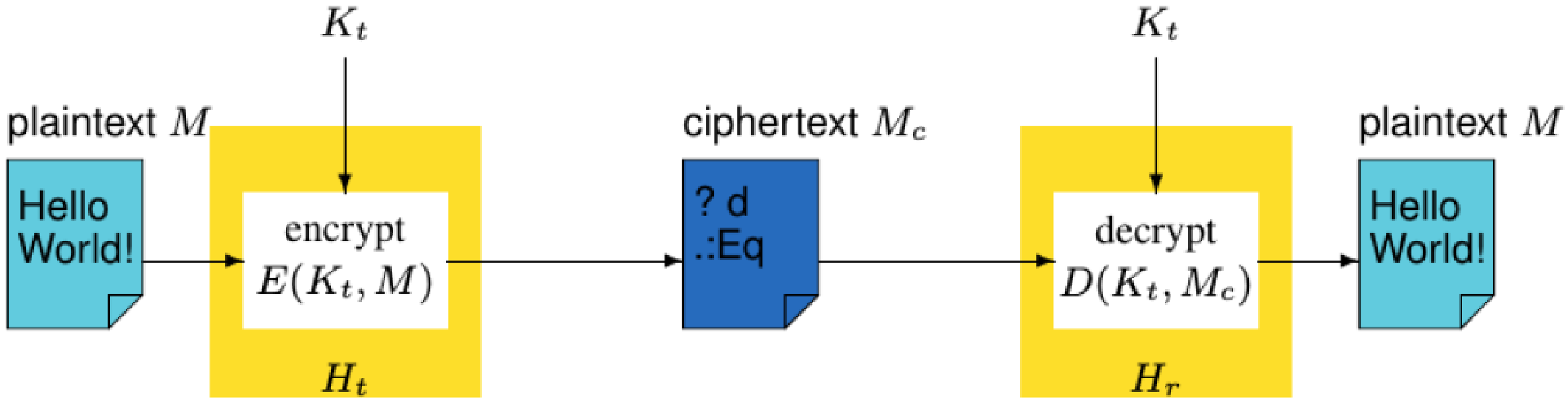
Encryption Properties



- Without K^{-1}
  - given M_c: can only find M by enumerating all possible K^{-1}
  - *a.k.a. brute-force attack*
  - takes a *very long time** if the domain of K^{-1} is large
    - (*****N****othing an extremely powerful [quantum?]*** ***S****upercomputer/cluster* ***A****nd a high budget cannot eventually beat*)

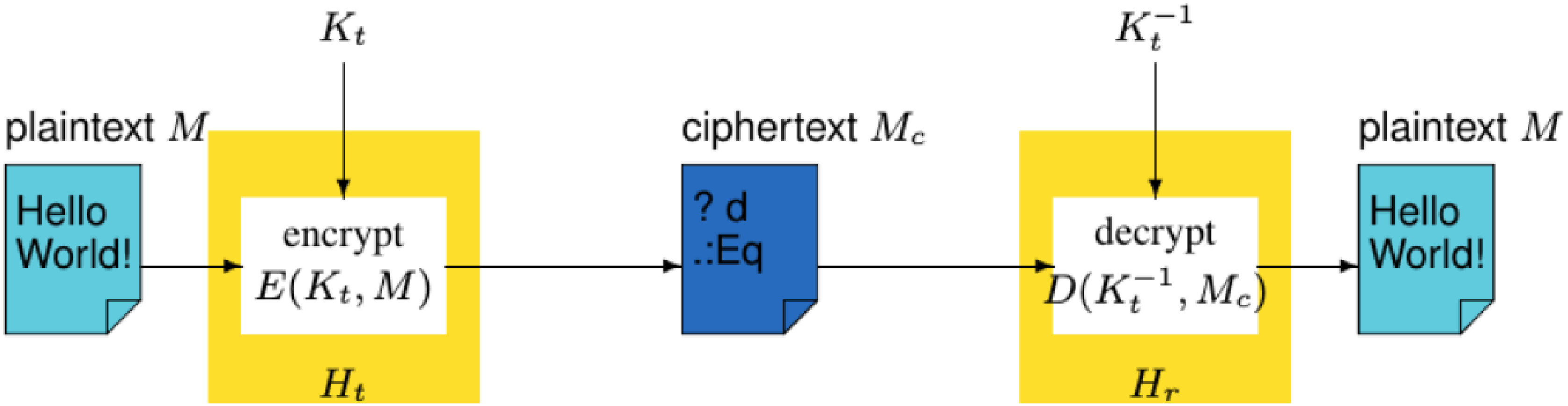
- Given M and M_c only
  - it should be difficult to obtain/guess the value of K *and* K^{-1}

Secret Key Encryption



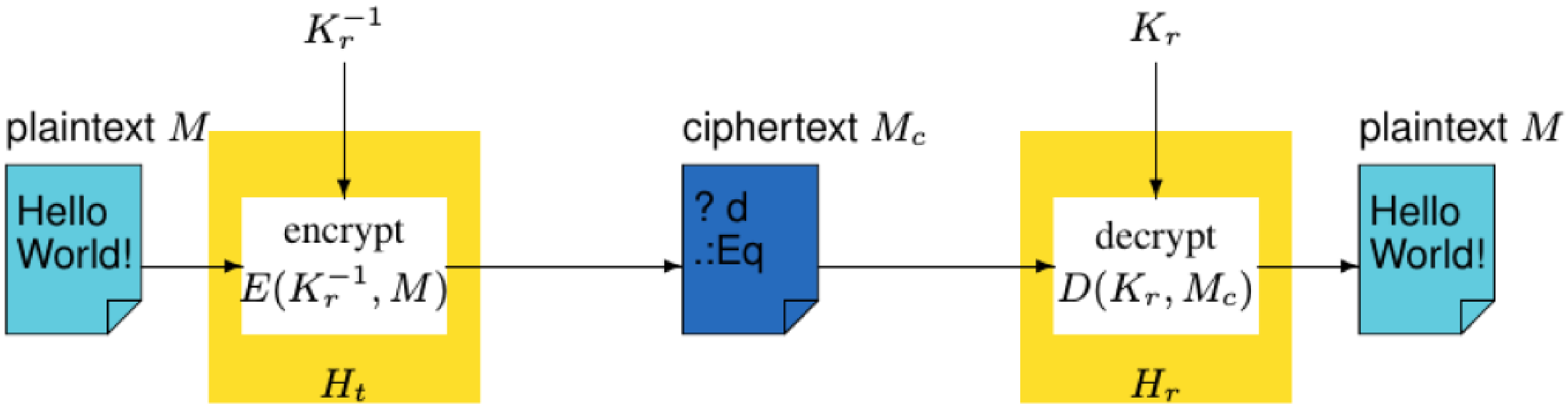
- K = K^{-1} (*e.g.* K_t *for both*)
- Also called ***Symmetric*** encryption
- Similar to
  - password-protecting a file
- K_t must be *carefully* distributed to all hosts who are to access the channel

Public Key Encryption



- $K \neq K^{-1}$
- Also called ***Asymmetric*** encryption (*public-private key pair*)
- K_t is called the **private-key** of host H_t and is only stored/used by that host
- K_t^{-1} is called the **public-key** of host H_t and is *freely* distributed
- Successfully decrypting messages => authenticated as coming from H_t
  - as only H_t could have encrypted the message with their corresponding private-key

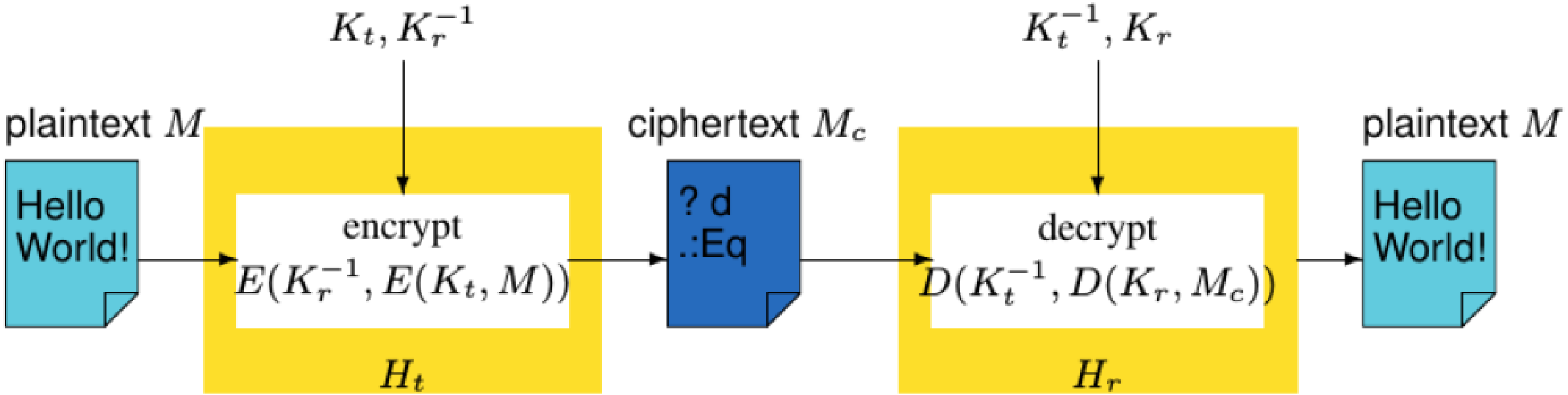
Public Key Encryption Confidentiality



- Encrypting a message with the public-key of the destination (K_r^{-1})
  - only H_r can decrypt the message

- Ensures confidentiality
  - (*as long as* H_r *keeps their* K_r *somewhere safe*)

Authentication and Confidentiality



- Encrypt/sign the message using your private key:
  - $E(K_t,M)$

- Encrypt the encrypted/signed message using the destination's public key and send it:
  - $E\ (K_r^{-1}\ , E(K_t,M))$

- Proof that only H_r may read it: $D\ (K_r\ , E(K_r^{-1}, E(K_t,M)))$ => $E(K_t,M)$

- Proof that only H_t could have sent it: $D\ (K_r^{-1}\ , E(K_t, M))$ => M

Encryption Comparison

- **Public Key**
  - Owner of private-key does not need to disclose its value
  - More secure
  - Slow encryption/decryption
  - Example: RSA

- **Secret Key**
  - Owner of secret/private-key needs to disclose it in order to communicate
  - Less secure
  - Faster encryption/decryption
  - Example: DES

- *You can of course combine these*
  - *e.g. GnuPG (gpg)*

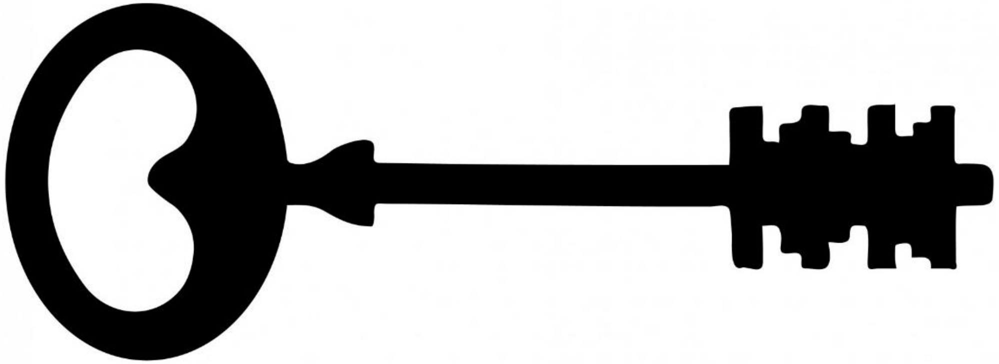
```
tar czvpf - filename.txt | gpg --symmetric --cipher-algo aes256 -o filename.tar.gz.gpg

gpg -d filename.tar.gz.gpg | tar xzvf -
```

Secure Channel Establishment

- *Obtain* the keys to use for encryption
    - How?

- If you require that all hosts must be given all the keys they may need *in advance*
    - it will not scale
    - it does not allow new hosts to be added

- Solution A: Agree on a **new key** right there and then
- Solution B: Use *trusted* **secure hosts**



Diffie-Hellman key exchange

- Secure key exchange, even over a public, unsecured, channel
- Whitfield Diffie & Martin Hellman
    - Original concept by Ralph Merkle
- Key can then be used for encryption by symmetric algorithms
- *Powerful/large systems can potentially defeat this*
- Used by SSL (*now TLS*)

Diffie-Hellman key exchange (cont'd)

- Bob and Alice agree on a public value **g** (*generator*) and a large prime number **p**

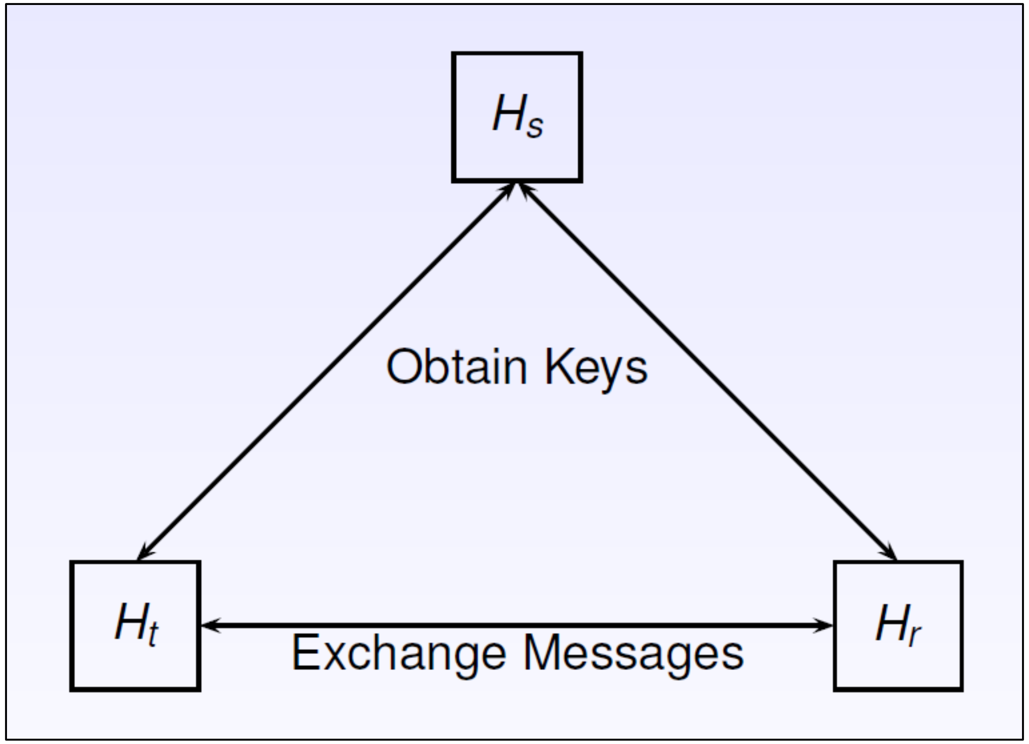
- Bob chooses a secret value **b** and Alice chooses a secret value **a**

- They each use their secret value to calculate their public value:
  - $x=(g^b \bmod p)$ for Bob and $y=(g^a \bmod p)$ for Alice
  - they exchange these (public) values

- They then use the other's public value to calculate the shared secret key:
  - Bob: $y^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p$
  - Alice: $x^a \bmod p = (g^b \bmod p)^a \bmod p = g^{ba} \bmod p$
  - this result/key can now be used by them to communicate securely

- *Eve, the eavesdropper, cannot guess/derive the shared secret key since she does not know either of the secret values* a & b

Secure Hosts: Key Server



- Each host knows how to communicate with the server securely
- The problem is how to (*safely*) obtain a key $\mathrm{K}_{\mathrm{t,r}}$ to talk to each other

Kerberos (Needham & Schroeder)

Imperial College London

- Kerberos is a user authentication system which knows your password
- It also knows the password of the user/resource you want to communicate with
- It's Key Distribution Center (KDC) can provide you with a "ticket"
  - allowing you communicate with that user/resource
- Tickets expire after a predefined amount of time
  - in which case you have to re-login to generate new tickets
- Original version was vulnerable to "replay" "Man-In-The-Middle" (MitM) attacks
  - these have now been addressed (*until the next issue is found*)
- Visualisation of Kerberos

Attacks on Cryptography Algorithms

- Algorithms rely on their being no known quick solution

- May not be true for long (*e.g. RSA1024, SHA-1 hash, MD5 hash, WEP, CSS, etc.*)

- Popular algorithms are constantly tested to identify potential issues/vulnerabilities
  - *before a "black-hat" finds one...*

- Quantum computing is expected to bring many changes to cryptography within the next decade

Logging & Auditing

- Your system is probably already keeping logs
  - Linux System Logs (`/var/log/`)
  - Windows Logs (*Event Viewer*)
- syslog protocol
  - Linux: syslog-ng
- Used to identify system or network issues
- Can also be used as evidence of criminal activities
- Logs have to be monitored/managed
- Anything useful identified by a log audit should be stored permanently

Next week: Practical Demonstration

- Remember to bring your laptop with **Wireshark** installed
  - (*wireless connection required*)

- You can also bring any other wireless device
  - to become the sender of messages
  - (*while others will be trying to read them*)

- This will take place right here in 311 as usual

Q&A

Imperial College London

- There is still an ongoing **assessed coursework** on the course website(s)
  - deadline: Wednesday 22/02/2017
    - *Worksheet #3 solutions will be uploaded at the end of this week*
- **Suggested reading**: Tanenbaum#8; Peterson#8; (Stallings#P9); Kurose#8.
- Please provide *anonymous* feedback on www.menti.com using the code **40 02 50**
  - *always active throughout the term*
- You can also provide *eponymous* feedback or ask questions via email (*username:* ***kgk***)
- Thank you for your attention
- **Movie of the week**: Takedown
- **Next time**: Practical Demonstration & the Future!

Acknowledgements

- *Many thanks to:*
  - *Anandha Gopalan*
  - *Peter McBrien*