

# **ITMOC ITO.012**

## *Session 9: Security*

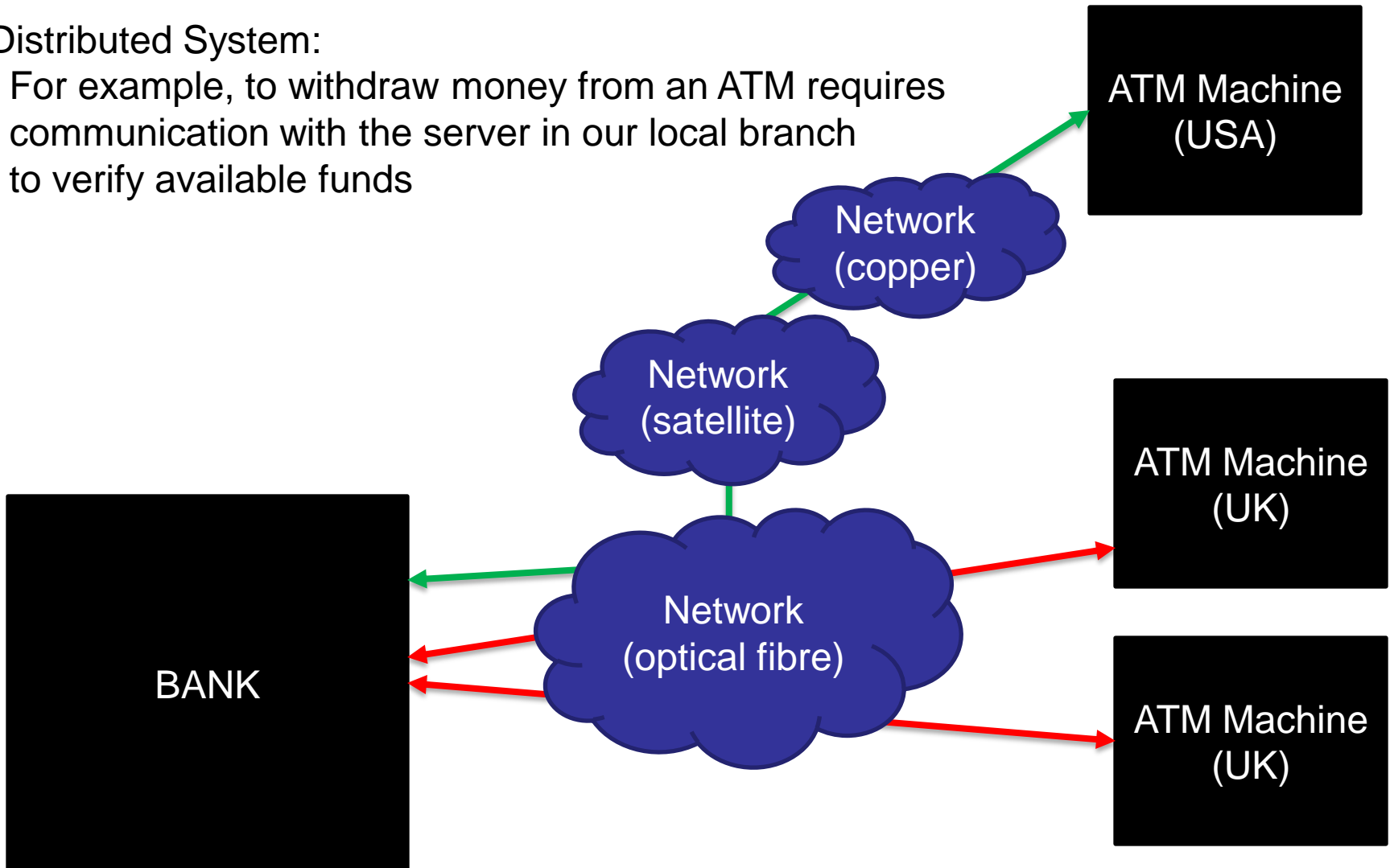
Dan Fitton  
df@comp.lancs.ac.uk

[http://www.comp.lancs.ac.uk/computing/staff/kc/keiths\\_teaching\\_ICT.html](http://www.comp.lancs.ac.uk/computing/staff/kc/keiths_teaching_ICT.html)

# Recap 1

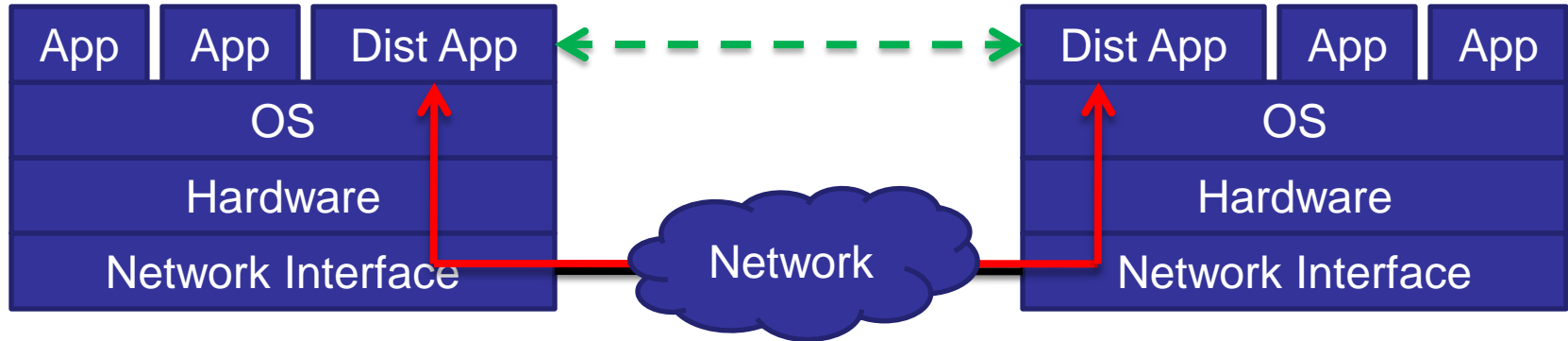
Distributed System:

For example, to withdraw money from an ATM requires communication with the server in our local branch to verify available funds



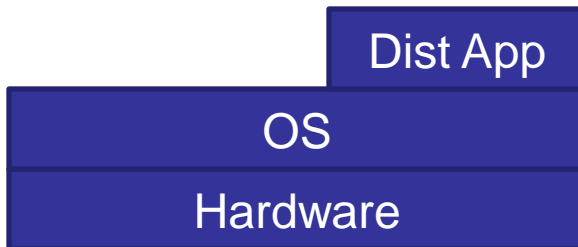
# Recap 2

## Distributed System



Addressing, unreliability, migration, extensibility etc

## Problem of Heterogeneity



Java, C, Visual Basic etc

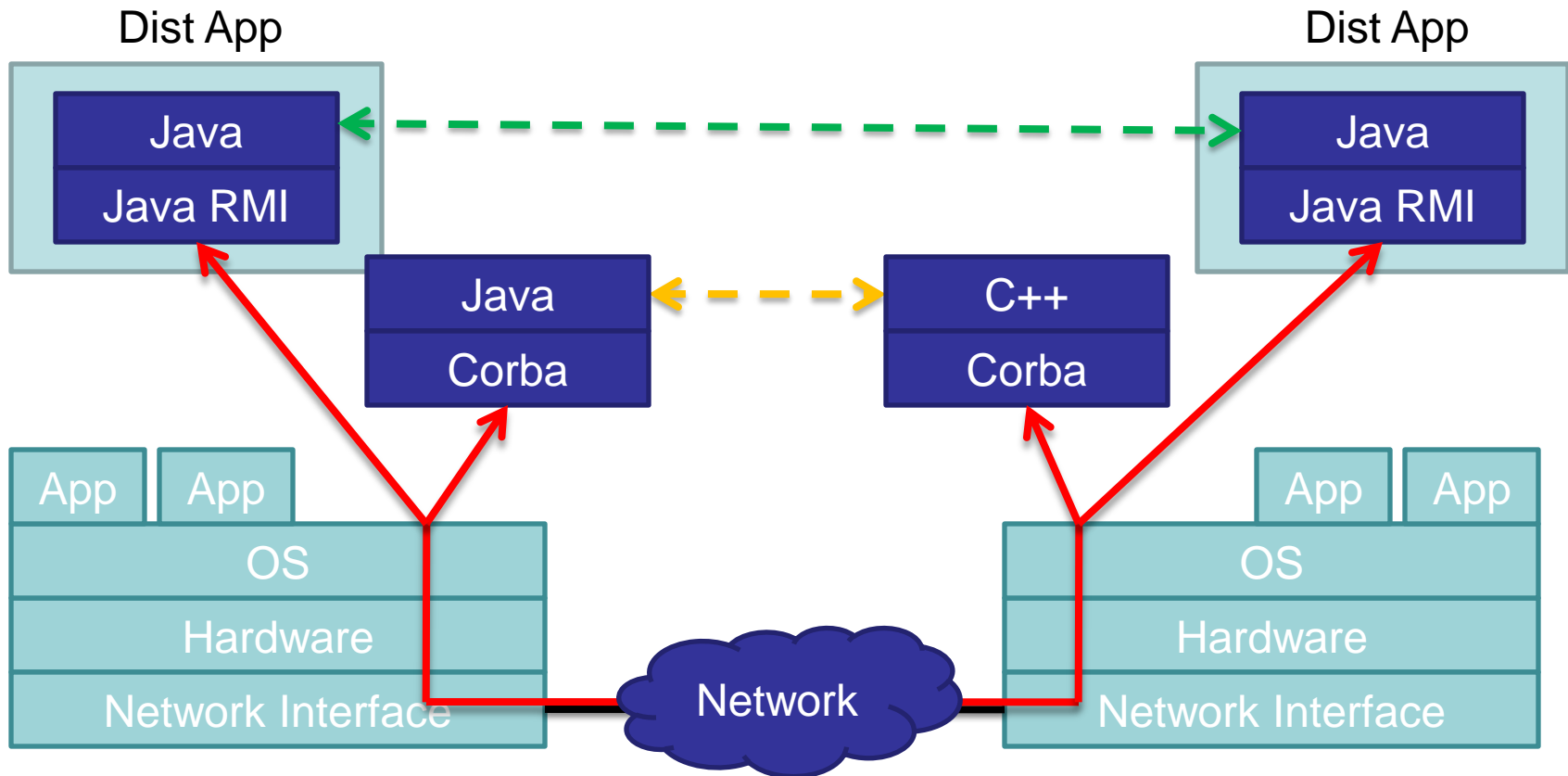
Linux, Windows, Solaris etc

x86, PowerPC, SPARC etc

# Recap 3

## Middleware:

Additional part of programming language to simplify the development of distributed applications by, for example, hiding the complexity of the network and provide facilities such as naming



# Security

- Overview of Lecture
  - Security problems
  - Principles of security
  - Basics of cryptography
    - Secret key encryption
    - Public key encryption
  - Authentication and key distribution
    - Needham and Schroeder protocol
- Additional reading
  - Tanenbaum chapter 8

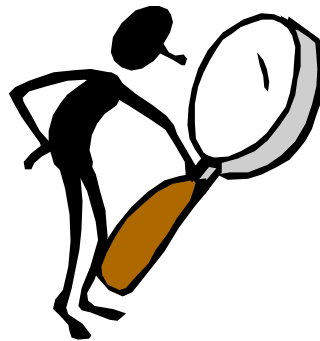
# Introducing Security

|  | <i>1965–75</i>   | <i>1975–89</i>  | <i>1990–99</i>                                | <i>Current</i>  |
|--|--|---|---|---|
| <i>Platforms</i>                       | Multi-user timesharing computers                                   | Distributed systems based on local networks                                 | The Internet, wide-area services              | The Internet + mobile devices                                 |
| <i>Shared resources</i>                | Memory, files  | Local services (e.g. NFS), local networks                                   | Email, web sites, Internet commerce           | Distributed objects, mobile code                              |
| <i>Security requirements</i>           | User identification and authentication                             | Protection of services  | Strong security for commercial transactions   | Access control for individual objects, secure mobile code     |
| <i>Security management environment</i> | Single authority, single authorization database (e.g. /etc/passwd) | Single authority, delegation, replicated authorization databases (e.g. NIS) | Many authorities, no network-wide authorities | Per-activity authorities, groups with shared responsibilities |

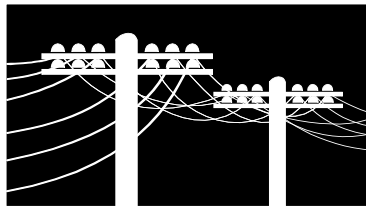
A historical perspective

# Security Vulnerabilities in Distributed Systems

## 1. Eavesdropping



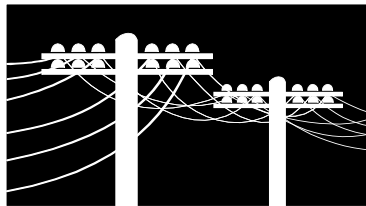
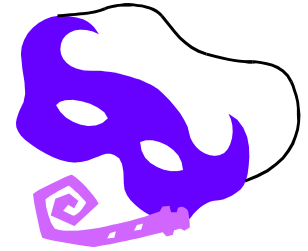
“Aha, I’ve just found your password”



# Security Vulnerabilities in Distributed Systems

## 2. Masquerading

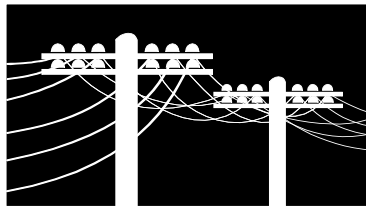
“I am amazon.com,  
honest, so please send  
me your credit card  
details”



# Security Vulnerabilities in Distributed Systems

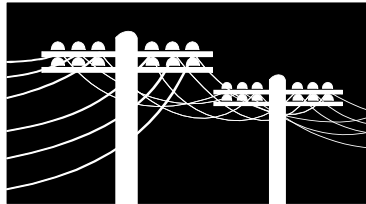
## 3. Tampering

“Please credit my account with £1m, not £1”



# Security Vulnerabilities in Distributed Systems

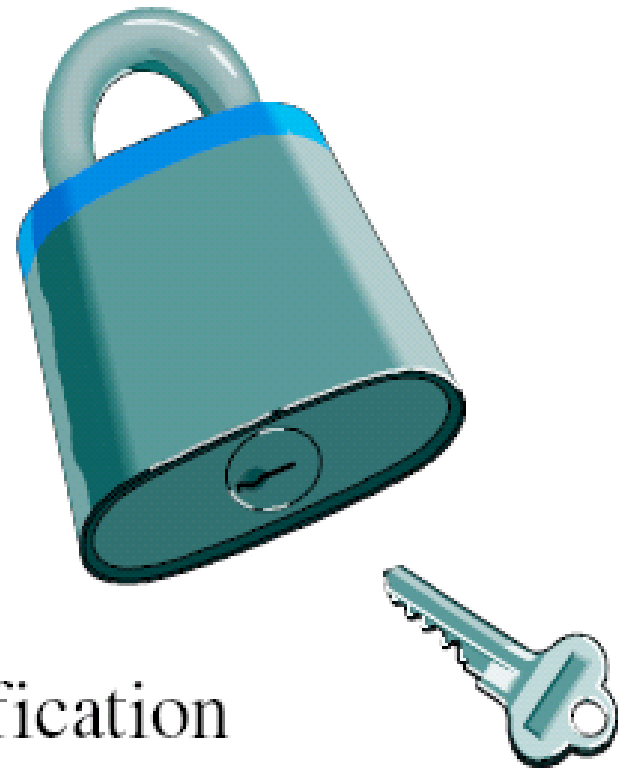
## 4. Replaying



“Lets have that one again”

# Security

- Authentication
- Cryptography
- Message Integrity
- Non-Repudiation
  
- Key distribution and Certification



# Security

- Laws differ from country to country
  - Many attacks mounted from countries with weaker legislation and few treaty agreements
  - Often a problem of jurisdiction – who do I complain to?
- Public domain software
  - Growing problem of freeware security attacks
  - Can no longer rely on their being a limited pool of security ‘experts’

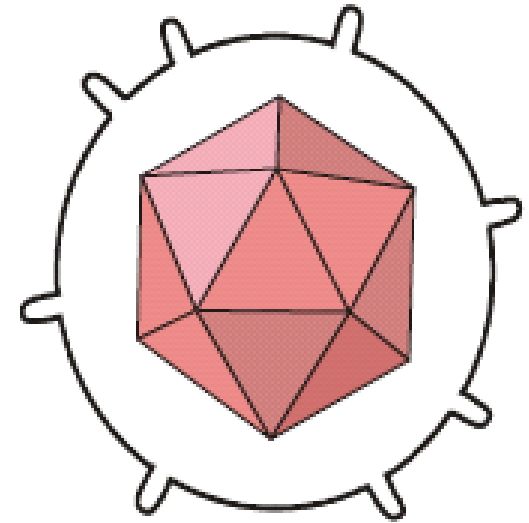
# Types of Attack

- Fraud
- Destructive – can also be achieved by ‘traditional means’
- Intellectual Property theft
- Brand theft
- Identity theft
- Privacy violations
  - Targeted attacks (stalking/ espionage/ national intelligence or spying)
  - Data harvesting
  - Surveillance and Traffic Analysis
- Legal – you can often persuade judge and jury of anything!

# Attackers

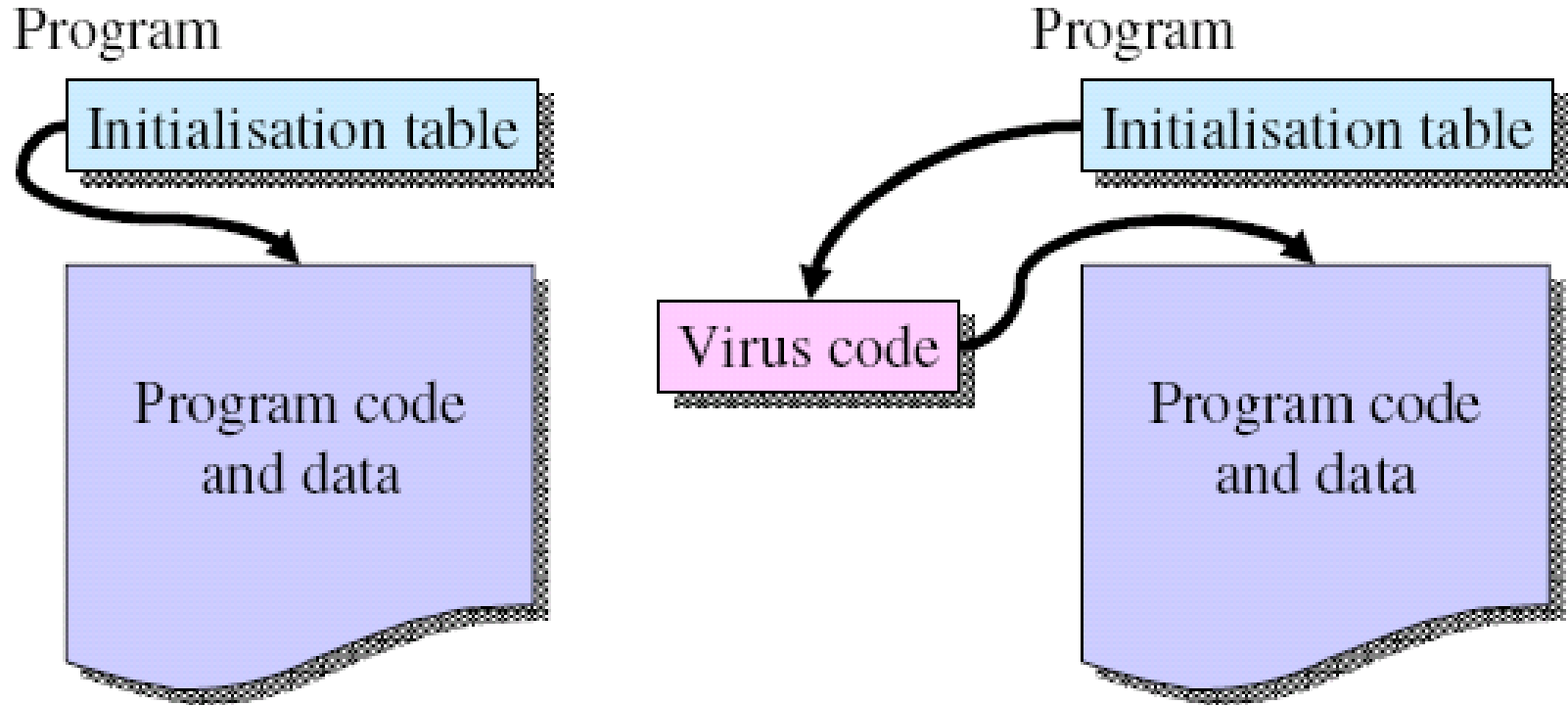
- The hacker
- Lone criminals
- Malicious insiders
- Industrial espionage
- Press
- Organised crime
- Terrorists
- Police
- National intelligence organisations
- ‘Info-warriors’

# Viruses



- Self propagating Trojan
- Source
  - Downloaded/ E-mailed files (MS Word macros)
  - Disc boot sector
- Use
  - Destroy data
  - Copy data
  - Update data

# Typical Virus



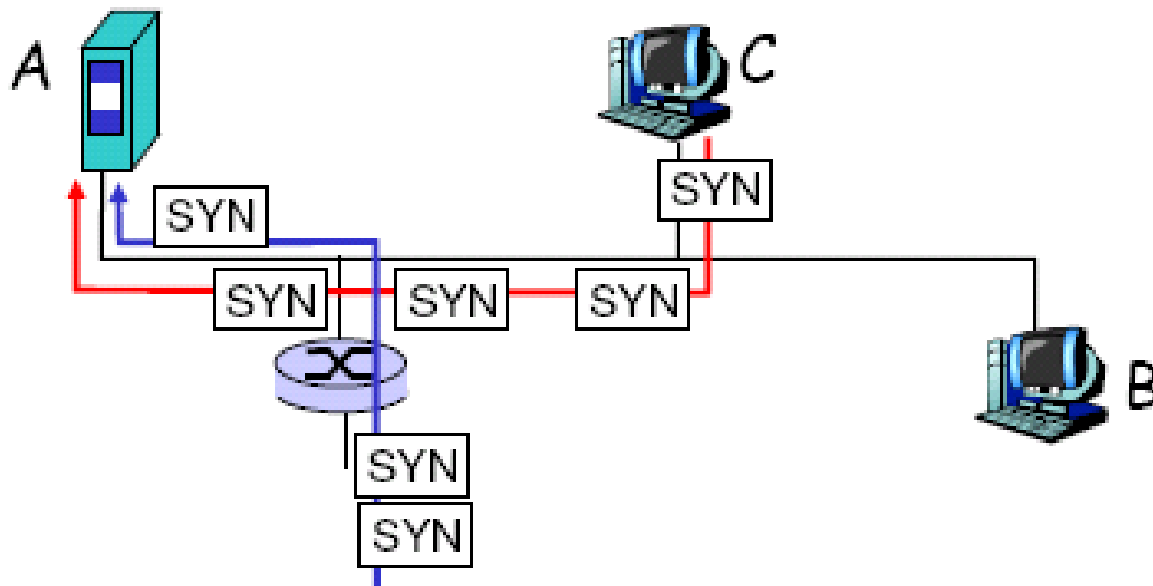
- Same idea for disk/ boot sector virus

# Worms

- Deploy segments of code around network
- To achieve
  - Malicious damage
  - Significant processing (poor mans multiprocessor)
- Often use buffer over-run attacks

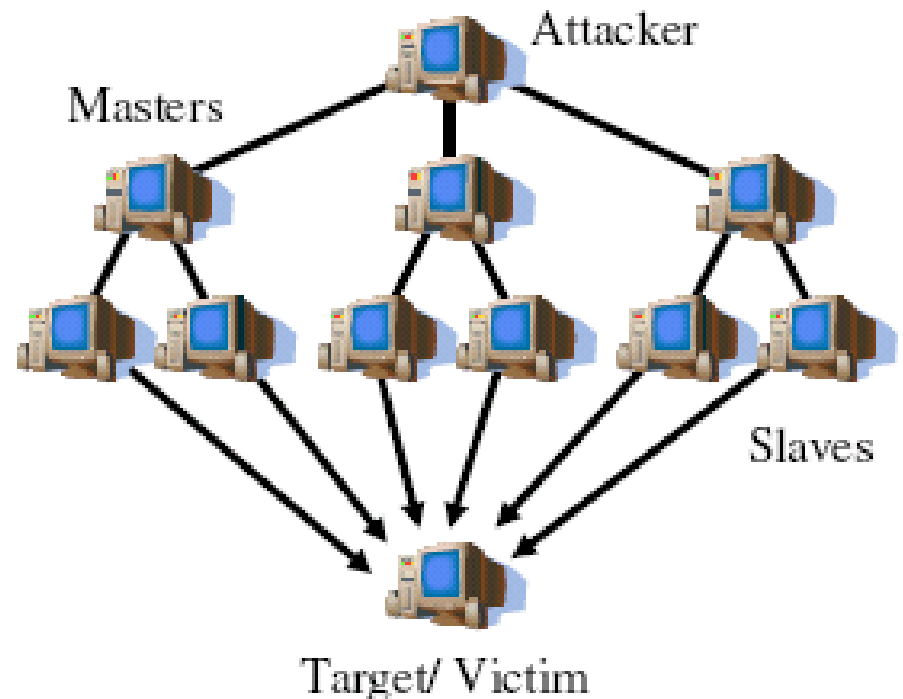
# Denial of Service Attack

- Distributed Denial of Service Attack



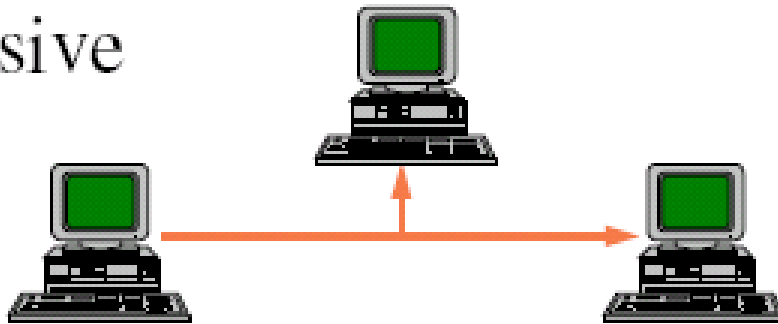
# DDoS Attacks

- *Master* and *Slave* machines infected with zombies
- *Attacker* coordinates *Masters*
- *Masters* coordinate *Slaves*
- *Slaves* mount attack on *Victim*
- Typically spoof source address
  - Attack cannot be traced
  - Victim cannot filter traffic or retaliate

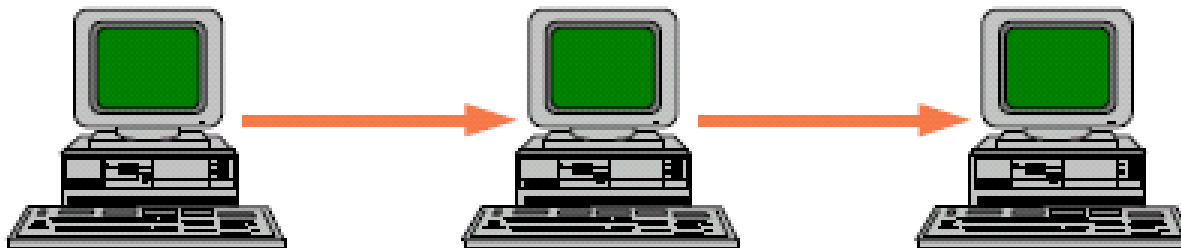


# Wire Tap / Sniffing

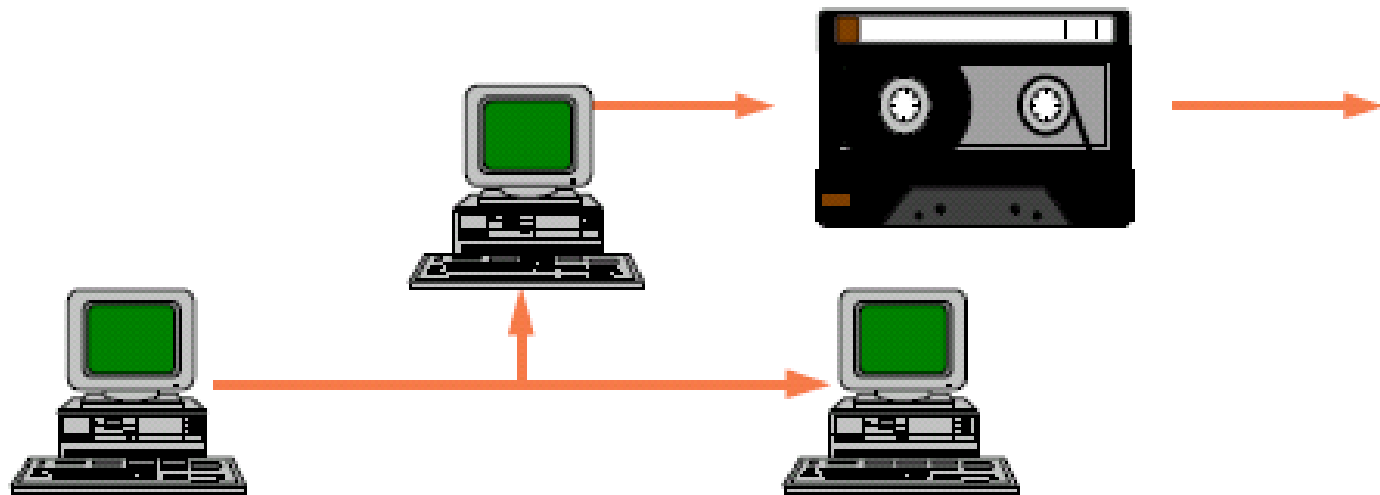
- Passive



- Active with Masquerading

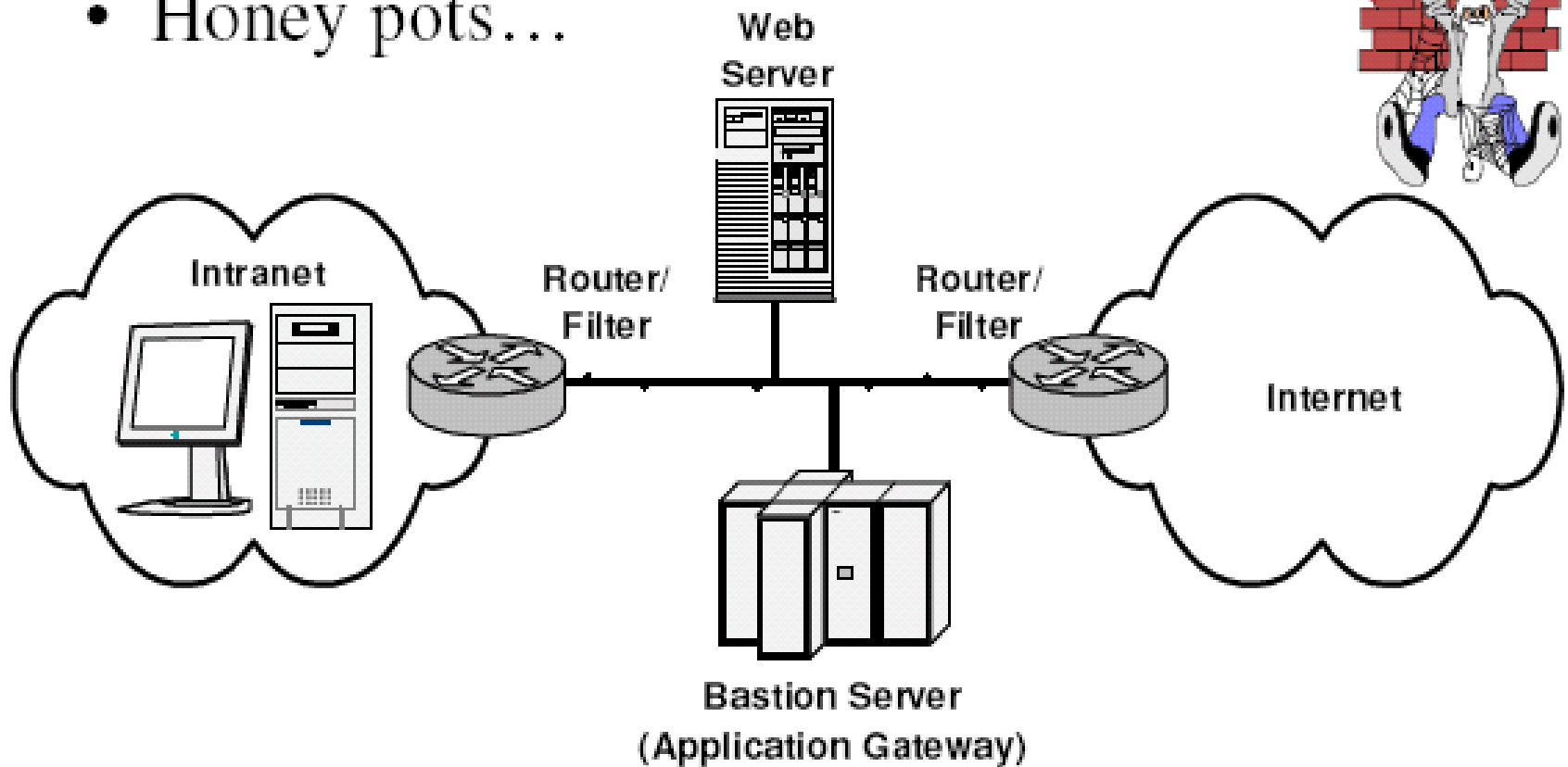


# Replay Attacks and Spoofing



# Firewalls and Intrusion Detection

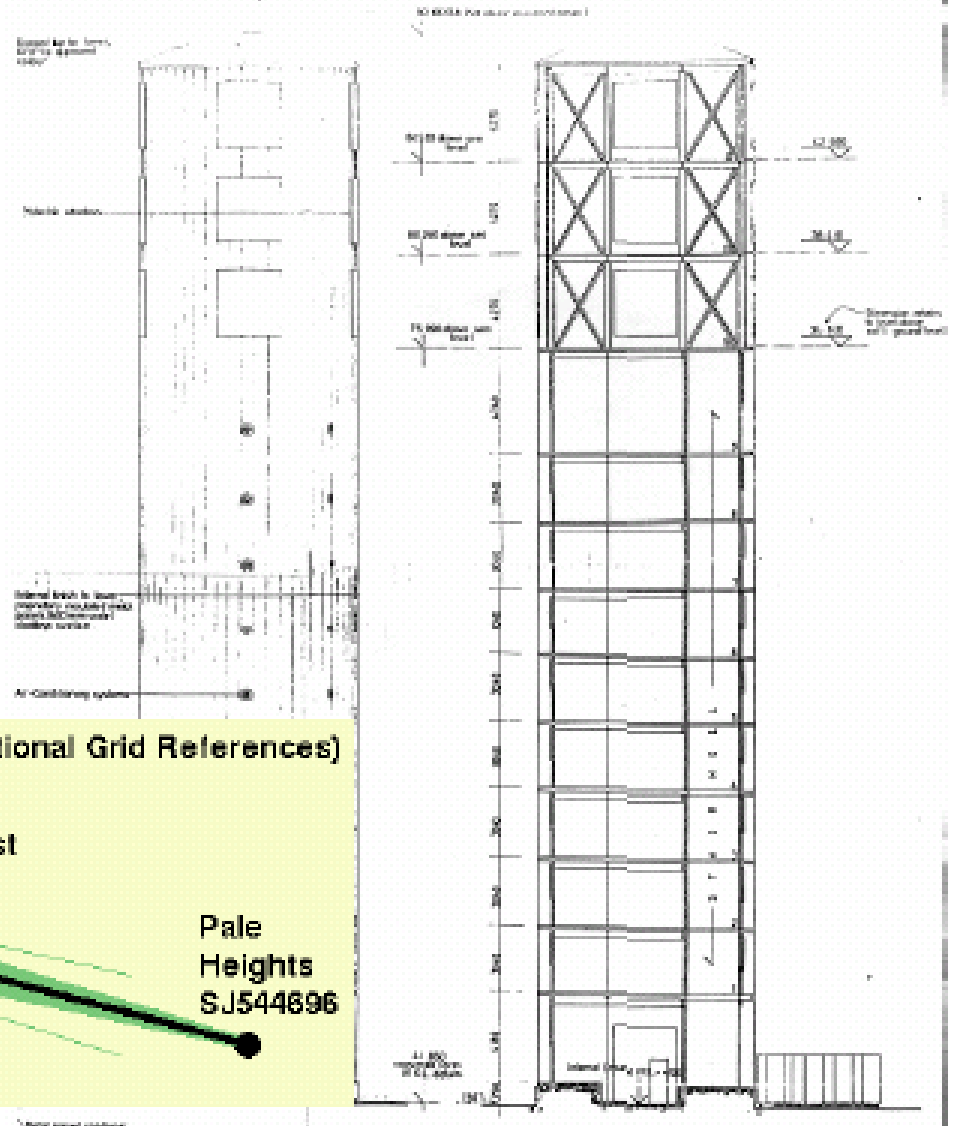
- Beware strange behaviour
- Honey pots...



# Mobile Security

- Radio networks are, by their nature, less secure than wired ones ...but that doesn't say that much!
- You find a wireless network...
  - Should the network trust you?
    - Traditional problem – just harder
  - Do you trust the network?
    - A more unusual problem

# Capenhurst Tower, Cheshire



Gwaenysgor SJ078818

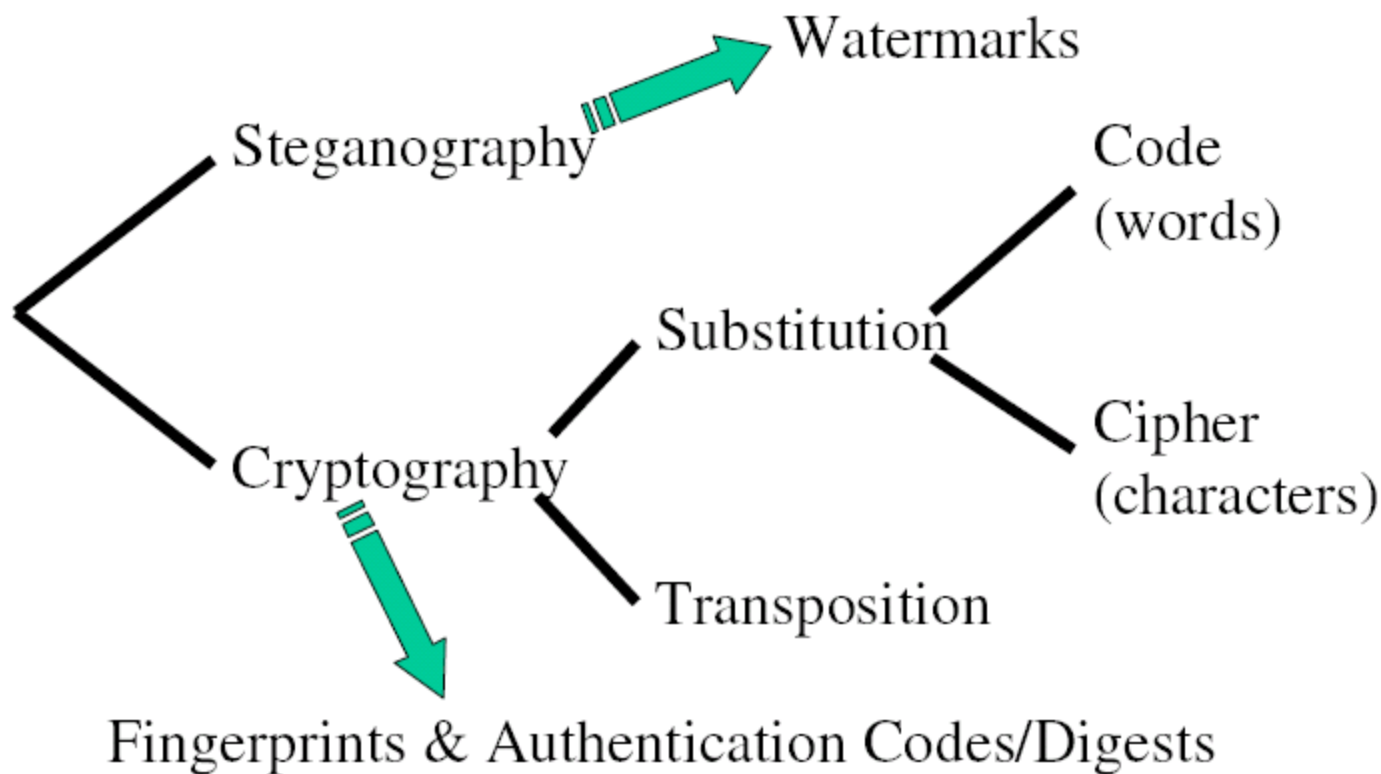
(Figures are National Grid References)

Capenhurst  
SJ364742

Pale  
Heights  
SJ544896

Capenhurst appears to sit about 100m south of the centre line of the beam, but is well within the BT 'beamwidth'.

# The “Secrecy Hierarchy”

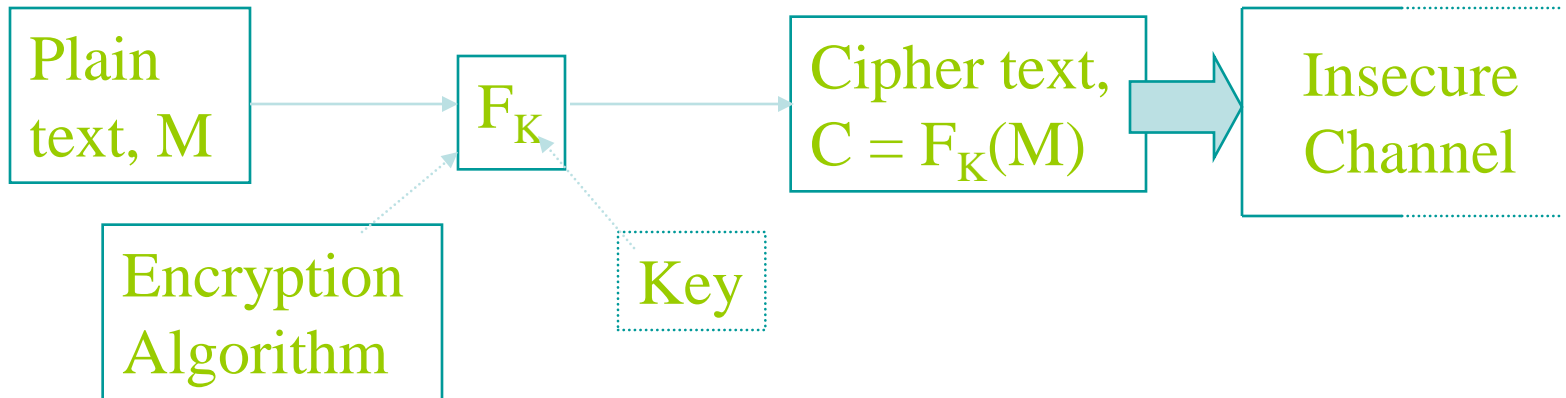


# Checksums and Message Digests

- Used to check message integrity
- Checksums widely used in Internet protocols
- Message Digests next level
  - Complex algorithms ‘ensure’ that no two messages can (should?) give same digest
  - Most widely used system MD5
    - Now known to be flawed, but still pretty safe

# Introduction to Cryptography

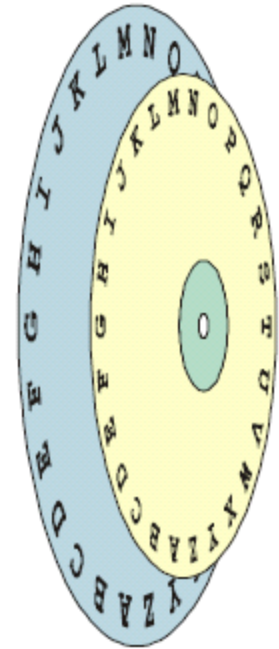
- To encrypt a message  $M$  with key  $k$  ...



- To decrypt the coded message  $C = F_K(M)$ ,
  - need the decryption key  $K'$ ,
  - perform the inverse process to recover  $M$ .

# Substitution Ciphers

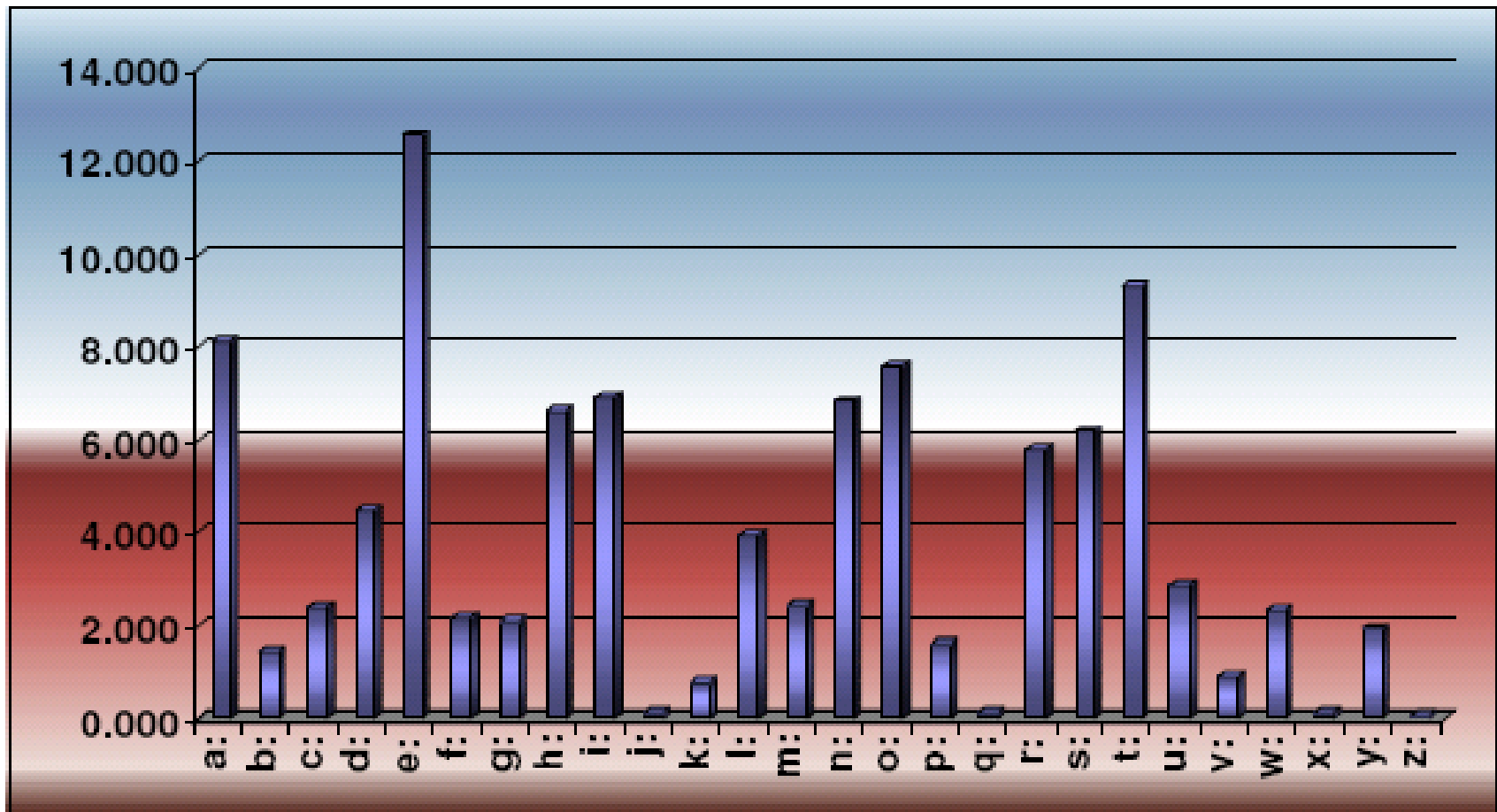
- Caesar Cipher
- Mono-alphabetic
  - Requires an offset or ‘key’
    - How much to rotate the disc, or...



|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| Z | Y | X | W | V | U | T | S | R | Q | P | O | N | M | L | K | J | I | H | G | F | E | D | C | B | A |

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | O | M | P | U | T | E | R | N | E | T | W | O | R | K |
| X | L | N | K | F | G | V | I | M | V | G | D | L | I | P |

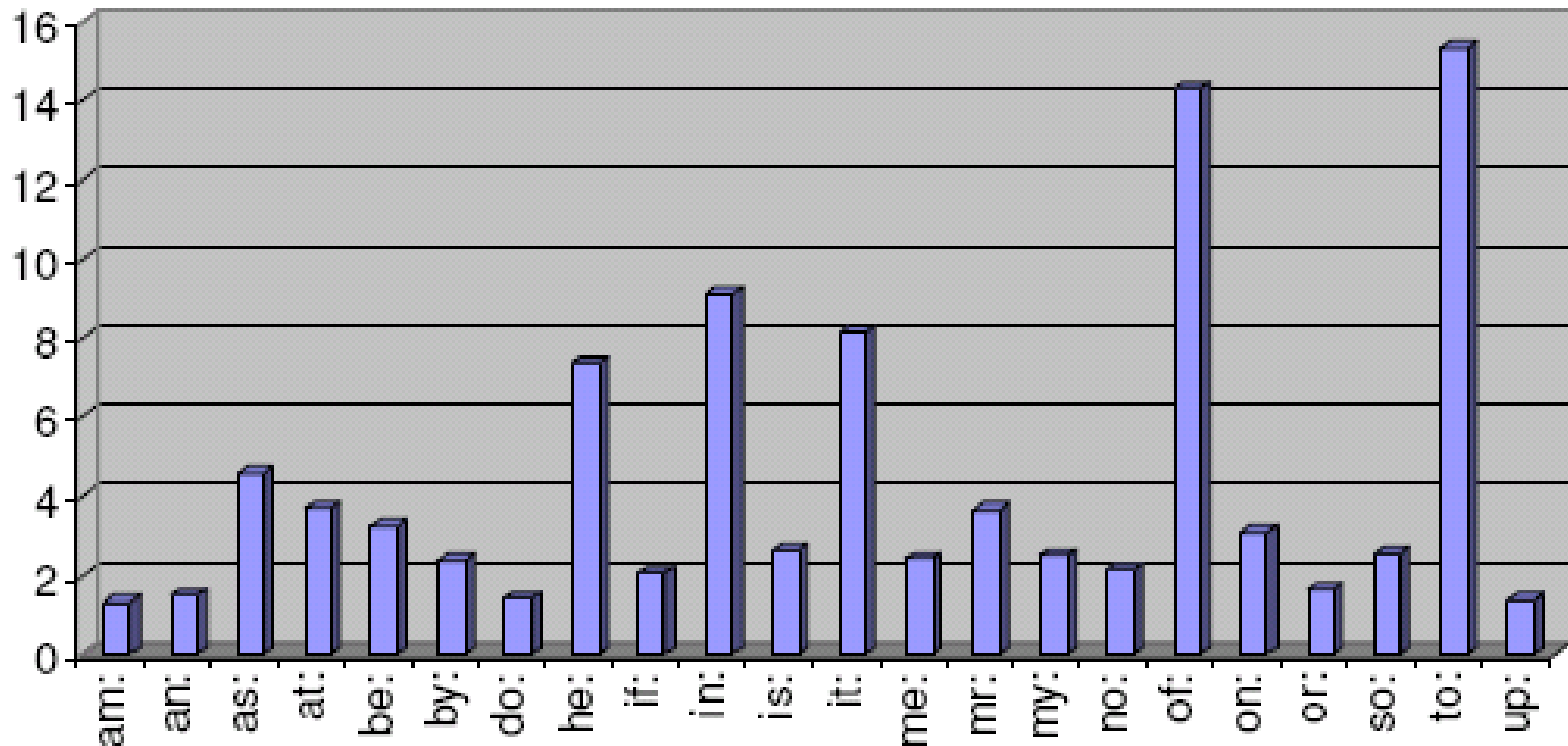
# Average Character Usage



# Digrams

## Digrams -- Little Dorrit, Dickens

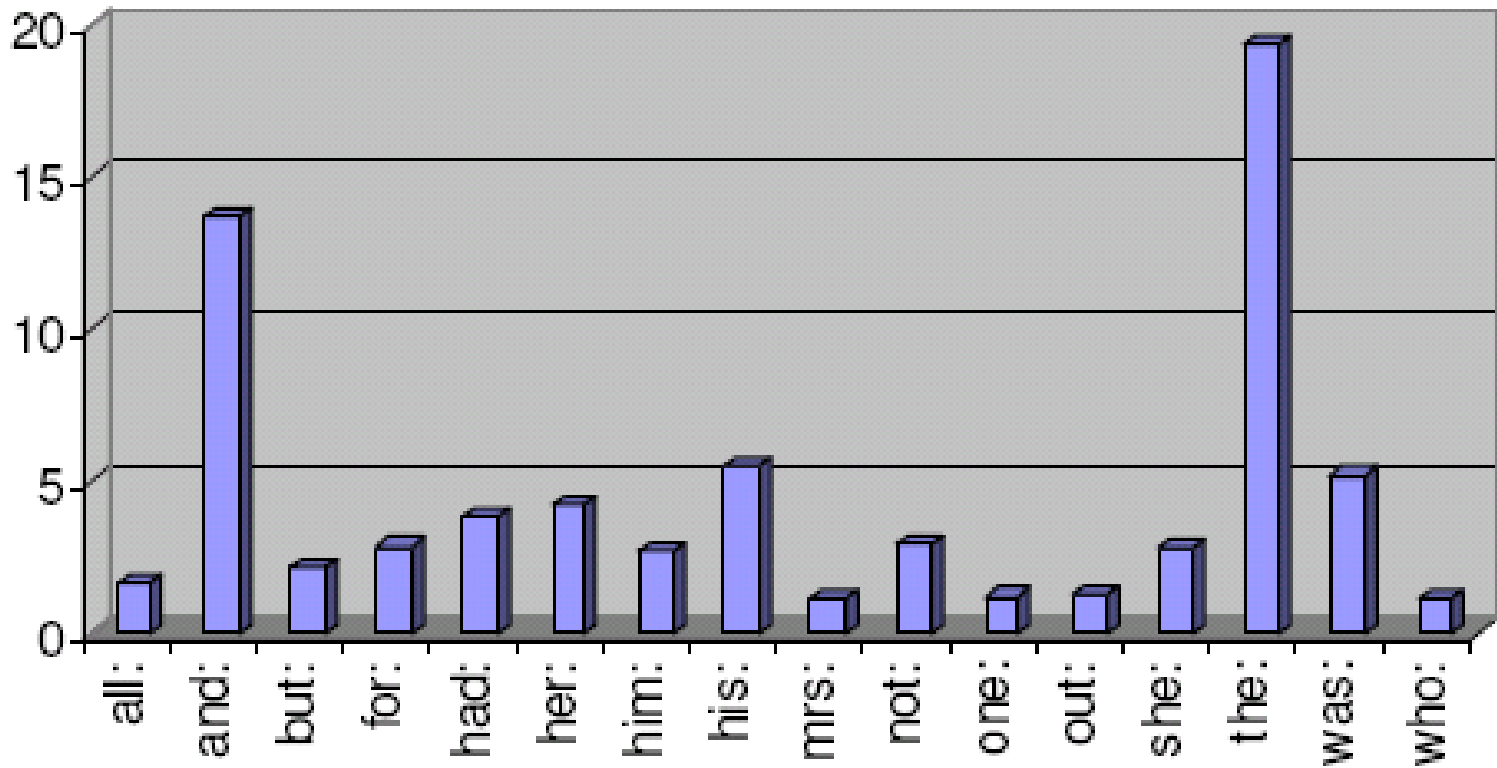
% (>1)



# Trigrams

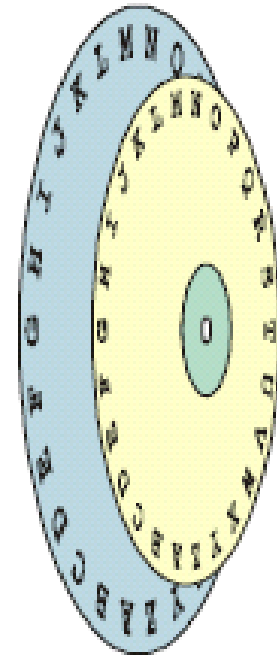
## Trigrams -- Little Dorrit, Dickens

% (>1)



# Substitution Ciphers

- Caesar Cipher
- Mono-alphabetic
- “Vigenère” Cipher
- Poly-alphabetic
  - Requires a code word
    - Which alphabet to use for each character



| A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|
| B | C | D | E | F | G | H | I | J | K |
| C | D | E | F | G | H | I | J | K | L |
| D | E | F | G | H | I | J | K | L | M |
| E | F | G | H | I | J | K | L | M | N |
| F | G | H | I | J | K | L | M | N | O |
| G | H | I | J | K | L | M | N | O | P |

# Poly-alphabetic Example

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|   | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | Z | Y | X | W | V | U | T | S | R | Q | P | O | N | M | L | K | J | I | H | G | F | E | D | C | B | A |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| I | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F | G | H | J | K | L | Z | X | C | V | B | N | M |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| T | M | N | B | V | C | X | Z | L | K | J | H | G | F | D | S | A | P | O | I | U | Y | T | R | E | W | Q |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| Z | G | H | J | K | L | Z | X | C | V | B | N | M | Q | W | E | R | T | Y | U | I | O | P | A | S | D | F |

Key →  
= ITA

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I | T | A | I | T | A | I | T | A | I | T | A | I | T | A |
| C | O | M | P | U | T | E | R | N | E | T | W | O | R | K |
| E | S | N | H | Y | G | T | O | M | T | U | D | G | O | P |

# Cracking Poly-alphabetic Systems

- Strength depends on codeword (key) length
- Look for common, relatively short words
  - They will appear often
  - Limited number of encrypted strings for these words
    - Number depends on length of key
  - Spacing between occurrences indicates key length
    - Use factors of this number to find possible lengths
    - And off you go ...using statistical analysis of letters again

# Transposition Ciphers

- Symmetrical
  - Sender and Receiver use same key
  - c.f. Public Key Cryptography (later)

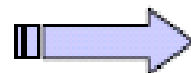
A N D R E W (Key)

1 4 2 5 3 6

C O M P U T

E R S E C U

R I T Y A B



|   |   |   |   |   |   |
|---|---|---|---|---|---|
| A | D | E | N | R | W |
| 1 | 2 | 3 | 4 | 5 | 6 |
| C | M | U | O | P | T |
| E | S | C | R | E | U |
| R | T | A | I | Y | B |

CERMSTUCAORIP EYTUB

# Secret Key Encryption

- A key,  $k$ , is only known to authorised people
  - a.k.a. *private key* encryption
  - Sender & receiver must share knowledge of  $k$ 
    - acquired through a secure channel
- The encryption function,  $F$  must also be known
  - This function need not be kept secret
- Commonly uses same key for encryption and decryption  $\Rightarrow$  *symmetrical* encryption

# Encryption/ Decryption Using Secret Keys

## Sender, A

1. acquire K
2.  $C = F_K(M)$
3. send C

C

## Receiver, B

1. acquire K
2. receive C
3.  $F^{-1}_K(C) = M$

K = key

M = message (plaintext)

C = ciphertext

F = encryption function

$F^{-1}$  = decryption function

# Using Secret Keys

- Encryption
  - Requires additional support to ensure both parties get access to the same secret key (*key distribution*)
- Authentication
  - Again, requires additional support in the form of often complex protocols to build up levels of trust

See Needham and Schroeder later

# Public Key Encryption

- First proposed by Diffie & Hellman (1976) to eliminate the need for trust
  - Encryption key,  $e$ , is made public
  - Decryption key,  $d$ , kept private (personal)
  - Relationship between  $e$  and  $d$  is a *one-way function*
- Based on product of 2 v. large primes ( $>10^{100}$ )
  - Determining prime factors of such a number is computationally intense & slow
- Asymmetrical encryption since  $e \neq d$

# Using Public Keys

- Encryption

Client

Client

Client

*Private communication*

Service

*Keep decrypt key  
( $K^{-1}$ ) private*

*Publish encrypt key ( $K$ )*

# Using Public Keys (continued)

- Authentication

Client

Client

Client

*Authenticated communication*

Service

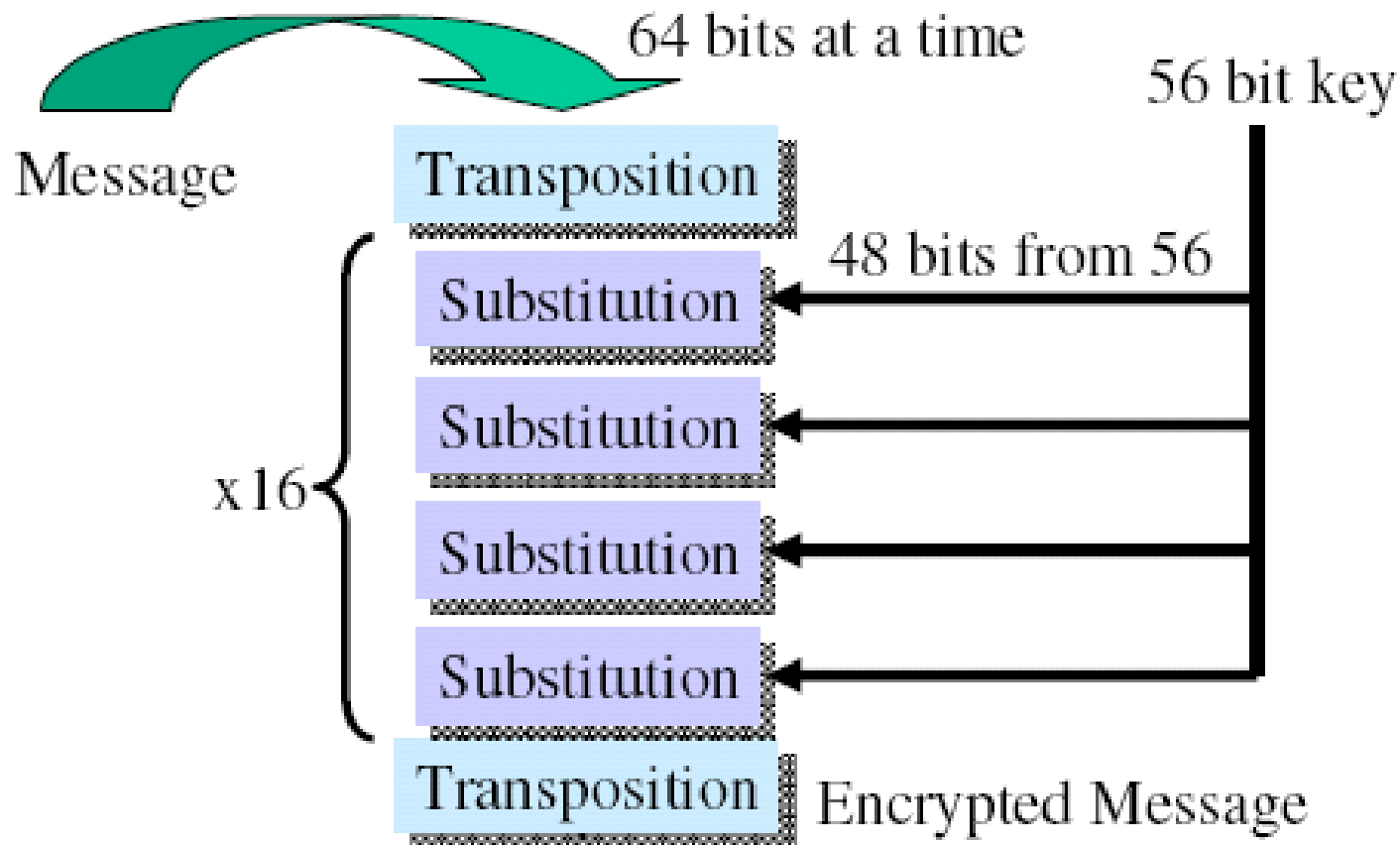
*Keep encrypt key  
( $K$ ) private*

*Publish decrypt key ( $K^{-1}$ )*

# Examples of Established Encryption Techniques

- **Data Encryption Standard (DES), 1977**
  - Secret key, based on sequence of substitutions and permutations (use controlled by US government)
  - Concerns over 56-bit key not being sufficient
  - Widely used, very fast, implemented in hardware
- **RSA algorithm (Rivest, Shamir, Adelman), 1978**
  - Public key, based on difficulty in finding factors of large numbers
  - No need to worry about distributing keys securely
  - Slower than DES
- Also PGP, AES, etc.....

# Data Encryption Standard



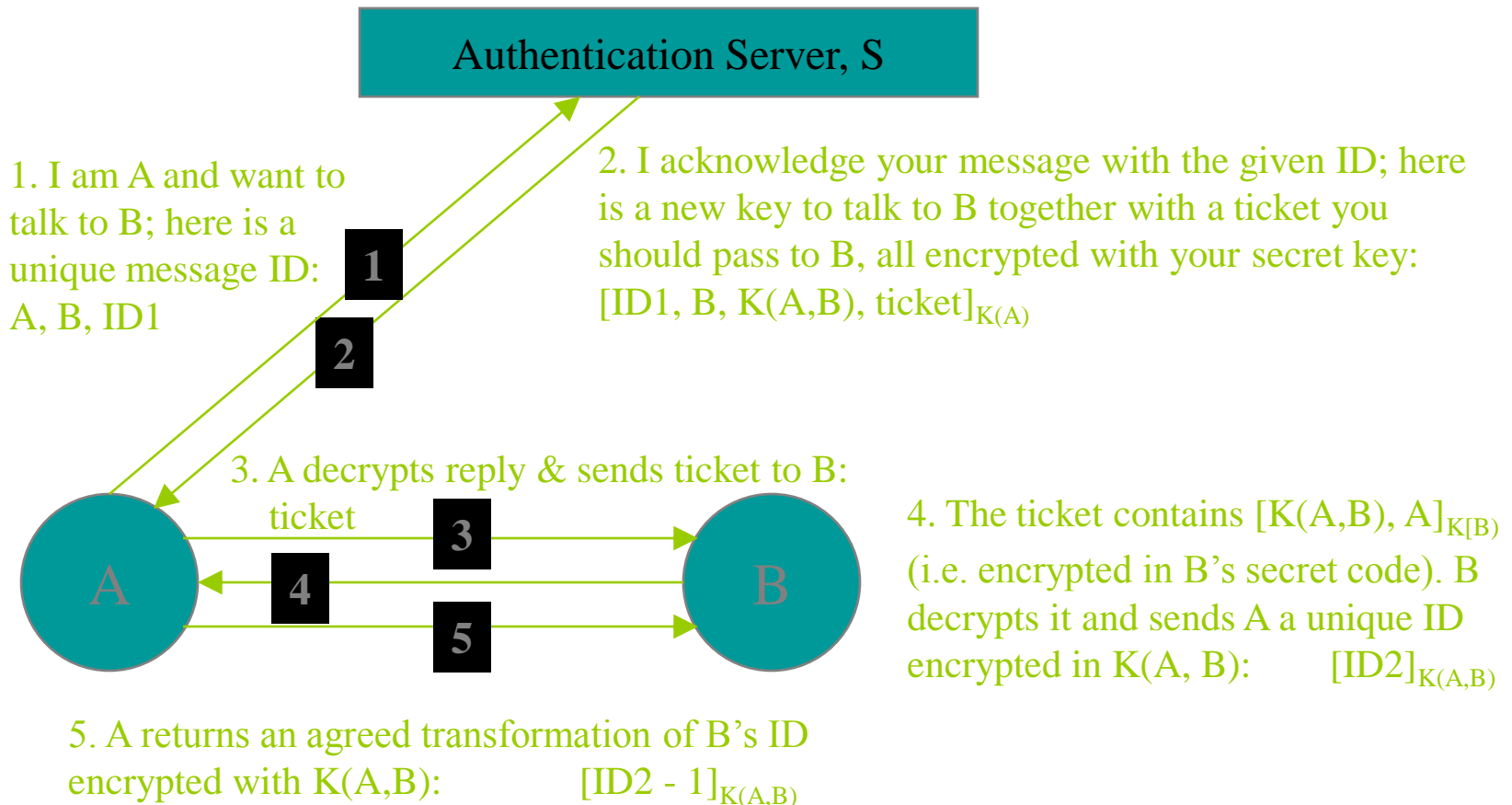
Why 56 bit key?

# Authentication & Secret Key Distribution

- Needham and Schroeder's protocol (1978)
  - Provides an *authentication server*
    - addresses the problems of authentication *and* secure key distribution
  - Two models
    - one based on secret keys
    - one based on public keys (not discussed here)

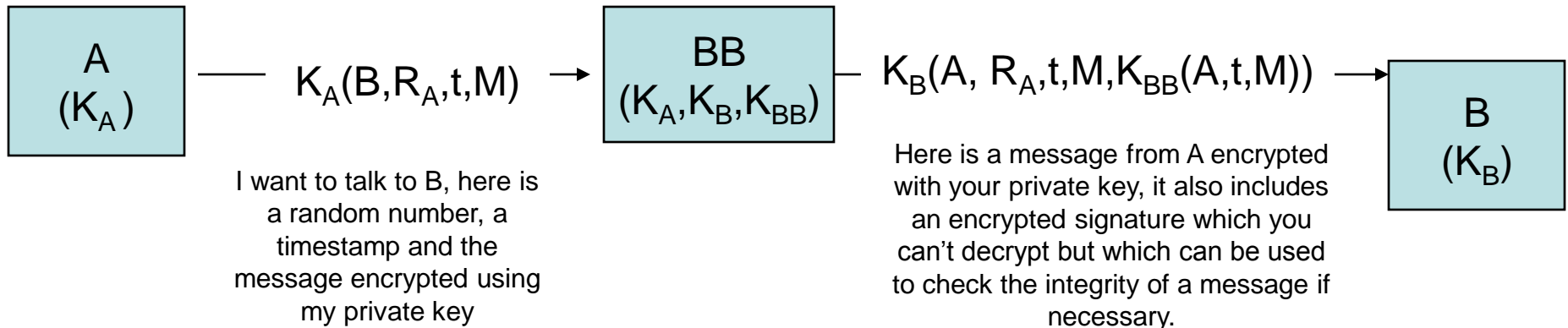


# Needham and Schroeder's Secret- Key Protocol



# Digital Signatures

- Analogous to a handwritten signature
  - Cannot be forged
  - Verify identity
  - Cannot repudiate messages
- Symmetric Key Signatures via a Central Authority



- Public Key Signatures
  - First use private key for encryption
  - Then use recipient's public key for encryption

# Certificates

- Secure public key distribution
- Public key published by a trusted central authority
  - You given them your public key
- Authority includes a signature which is the hash of the certificate signed with their private key
  - Calculate the hash and compare it to signature (decrypted with the authority's public key) to check integrity

# Summary

- Range of potential security vulnerabilities are inherent in distributed applications
- Range of potential security vulnerabilities when connecting computers to large networks
- Security problems are exacerbated in wireless networks
- Key mechanism is encryption
  - Various encryption algorithms exists
  - Concept of public key encryption
- Encryption used in a range of techniques to ensure security
  - Key distribution, digital signatures etc.