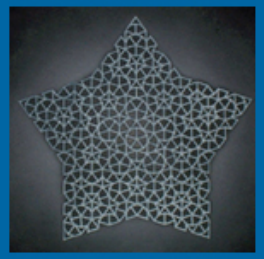
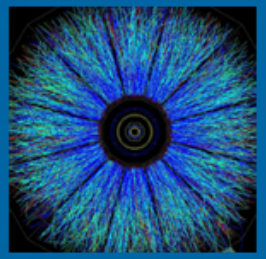
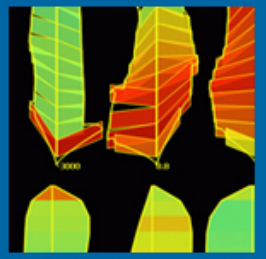
Swansea University
Prifysgol Abertawe

CS130:Professional Issues
Cryptography and Data Security

College of Science
Coleg Gwyddoniaeth

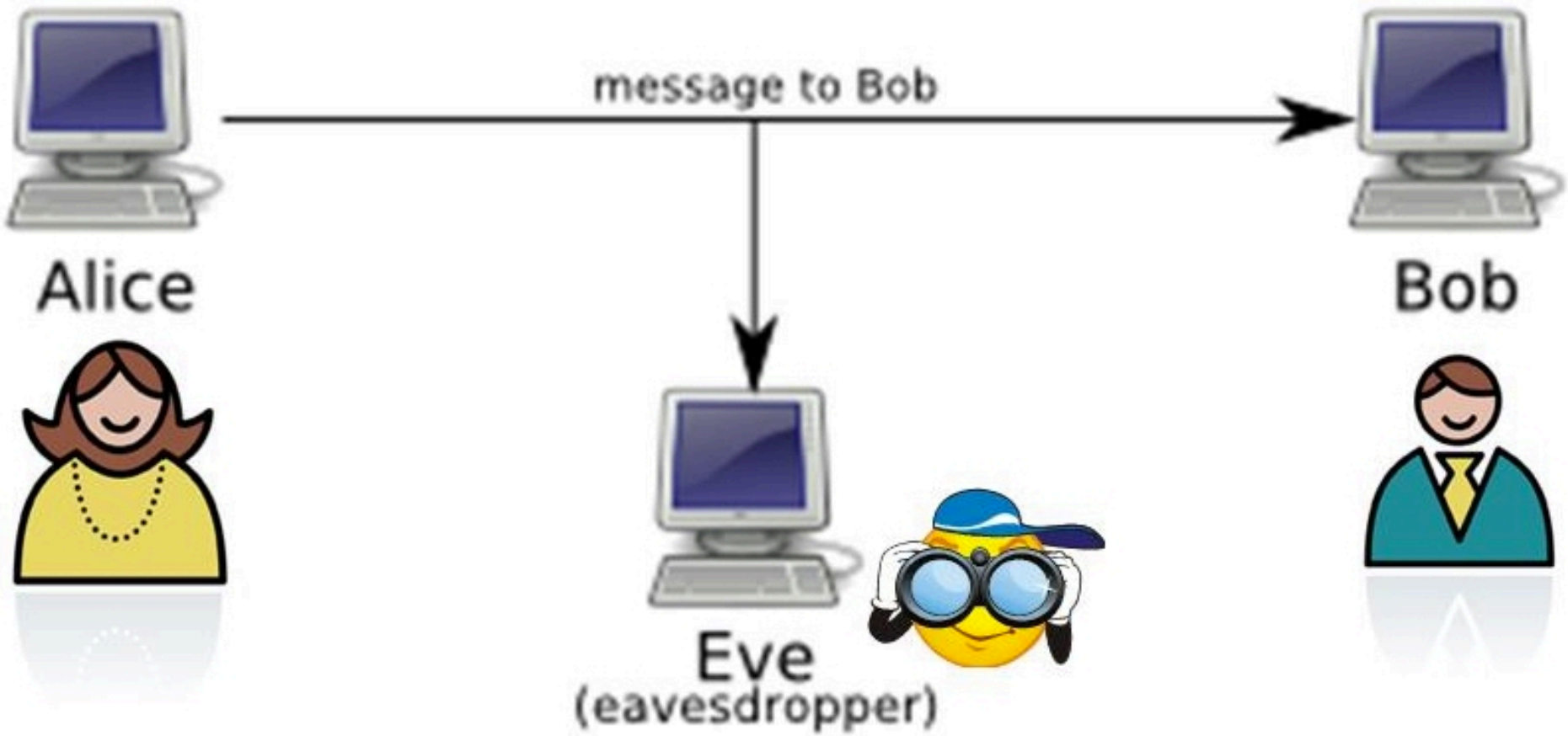
www.swansea.ac.uk/science

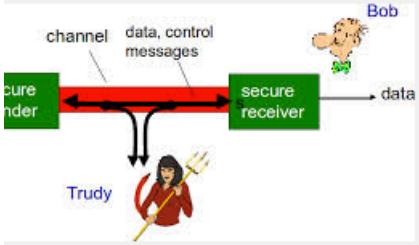
Topic Learning Goals

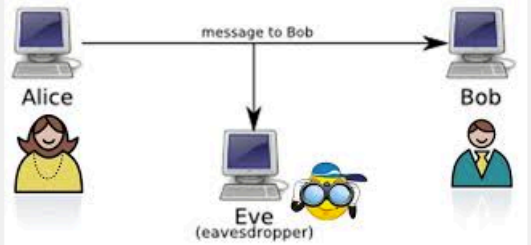
What key assumption must we make when transferring data between two secure terminals over a network?

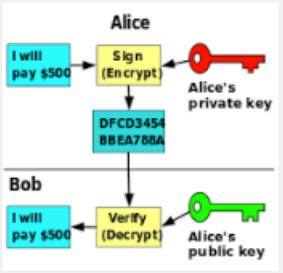
What vulnerability do we need to address with any encryption?

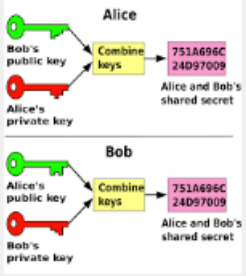
What is Kerchkoffs's principle and why is it superior to security through obscurity?


message to Bob
Alice
Bob
Eve
(eavesdropper)

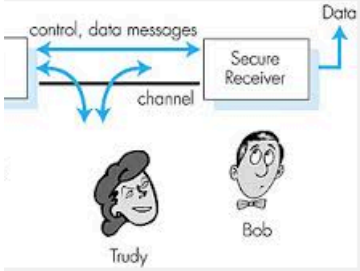

channel
data, control messages
Bob
secure receiver
data
Trudy

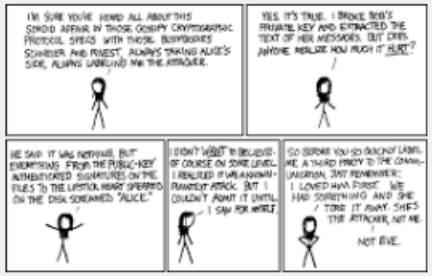

message to Bob
Alice
Bob
Eve
(eavesdropper)

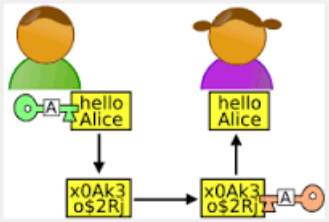

Alice
I will pay $500
Sign (Encrypt)
Alice's private key
DFCD3454 BBEA788A
Bob
I will pay $500
Verify (Decrypt)
Alice's public key

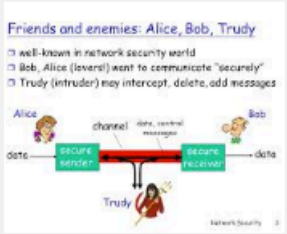

Alice
Bob's public key
Combine keys
751A696C 24D97009
Alice and Bob's shared secret
Alice's private key
Bob
Alice's public key
Combine keys
751A696C 24D97009
Alice and Bob's shared secret
Bob's private key

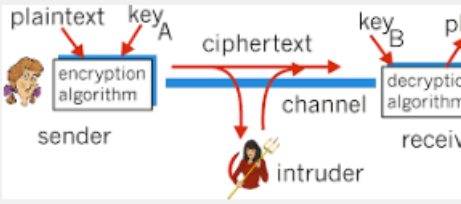

Message Integrity
• Alice is sending a message to Bob.
• Is Bob receiving exactly what Alice is sending?
ALICE
BOB
TRUDY


control, data messages
Data
Secure Receiver
channel
Trudy
Bob

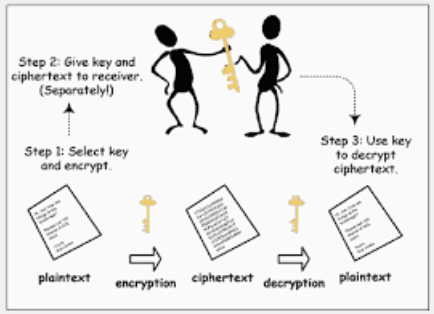

I'M SURE YOU'VE HEARD ALL ABOUT THIS SORDID AFFAIR IN THOSE GOSSIPY CRYPTOGRAPHIC PROTOCOL SPECS WITH THOSE BUSYBODIES ALICE, BOB, EVE, AND TRUDY. ALWAYS TAKING ALICE'S SIDE, ALWAYS LABELING ME THE ATTACKER.
YES. IT'S TRUE. I BROKE BOB'S PRIVATE KEY AND EXTRACTED THE TEXT OF HER MESSAGES. BUT DOES ANYONE REALIZE HOW MUCH IT HURT?
HE SAID IT WAS NOTHING, BUT EVERYONE'S FROM THE PUBLIC-KEY AUTHENTICATED SIGNATURES ON THE FILES TO THE LIPSTICK SMEARED ON THE DISK SCREAMED "ALICE."
I DON'T WANT TO BELIEVE. OF COURSE ON SOME LEVEL, I REALIZED IT WAS A KNOWN-PLAINTEXT ATTACK. BUT I COULDN'T ADMIT IT UNTIL I SAW FOR MYSELF.
SO BEFORE YOU GO QUICKLY LABEL ME A THIRD PARTY TO THE COMMUNICATION, JUST REMEMBER: I LOVED HER FIRST. WE HAD SOMETHING AND SHE TOOK IT AWAY. SHE'S THE ATTACKER, NOT ME.
NOT EVE.

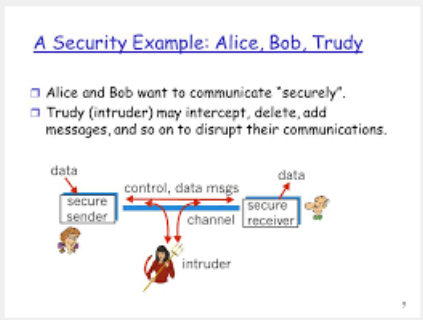

A
hello Alice
hello Alice
x0Ak3 o$2Rj
x0Ak3 o$2Rj
A

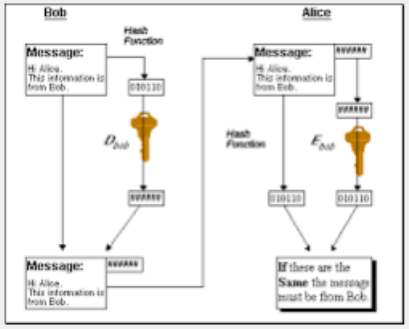

Friends and enemies: Alice, Bob, Trudy
well-known in network security world
Bob, Alice (lovers!) want to communicate "securely"
Trudy (intruder) may intercept, delete, add messages
Alice
Bob
channel
data
secure sender
secure receiver
Trudy

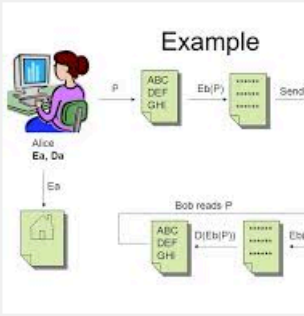

plaintext
key A
encryption algorithm
ciphertext
key B
decryptio algorithm
channel
sender
intruder
receiv


Bob
Eve


Step 2: Give key and ciphertext to receiver. (Separately!)
Step 1: Select key and encrypt.
Step 3: Use key to decrypt ciphertext.
plaintext
encryption
ciphertext
decryption
plaintext


A Security Example: Alice, Bob, Trudy
Alice and Bob want to communicate "securely".
Trudy (intruder) may intercept, delete, add messages, and so on to disrupt their communications.
data
secure sender
control, data msgs
channel
data
secure receiver
intruder

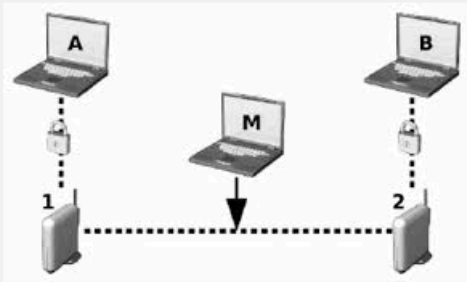

Bob
Alice
Hash Function
Message:
Hi Alice. This information is from Bob.
Message:
Message:
If these are the Same the message must be from Bob


Example
Alice
Ea, Da
Eb
Bob reads P

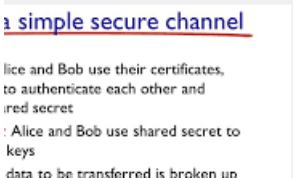

Privacy (PGP)

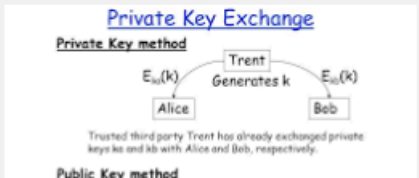

ALICE SENDS A MESSAGE TO BOB SAYING TO MEET HER SOMEWHERE.
UH HUH.
BUT EVE SEES IT, TOO, AND GOES TO THE PLACE.
WITH YOU SO FAR.
BOB IS DELAYED AND ALICE AND EVE MEET.
YEAH?
I'VE DISCOVERED A WAY TO GET COMPUTER SCIENTISTS TO LISTEN TO ANY BORING STORY.

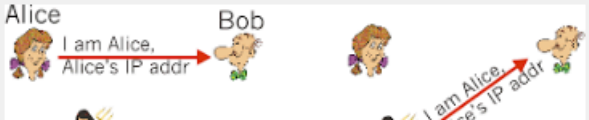

Plaintext
Encrypt
Cyphertext
Decrypt
Plaintext
Alice
Key
Eve
Key
Bob
Meet Alice & Bob


A
B
M
1
2


Alice


a simple secure channel
lice and Bob use their certificates, to authenticate each other and ired secret
: Alice and Bob use shared secret to keys
data to be transferred is broken up


Private Key Exchange
Private Key method
Trent
Generates k
Alice
Bob
Trusted third party Trent has already exchanged private keys ka and kb with Alice and Bob, respectively.
Public Key method


Alice
Bob
I am Alice, Alice's IP addr


Cryptology Basic Principles
• Alice – Bob – Eve

Who is The Man In The Middle?

When we send data across a network outside our physical control (untrusted Ethernet, all WiFi, all Mobile Data, The Internet) we must assume that a man in the middle can…

- **View** the messages content so read personal information in them
- **Intercept** the messages to stop them reaching the intended recipient
- **Repeat** the messages to try to gain access to a secure system

Cryptography – Classic Approaches

How do we address this problem? Our ideas have evolved quite a lot over time

Cryptography – a form of secret writing, any technique to disguise the meaning of a word to those who don't know how to interpret it

Transposition cyphers – hello world = ehlol owrdl

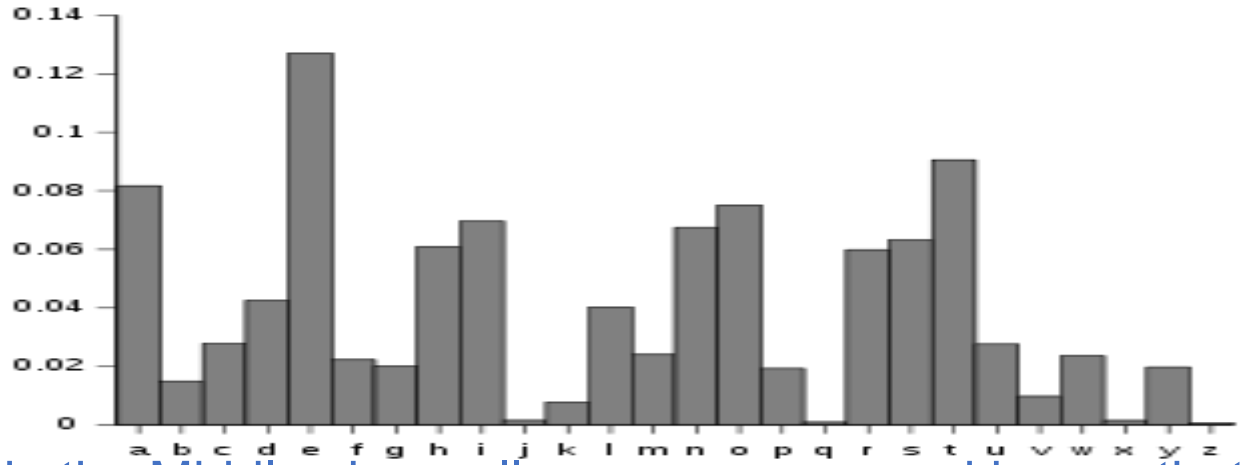
- Swap the ordering of letters around in some fixed pattern

Substitution Cyphers – hello world = ifmmp xpsme

- Take a letter and replace it with another letter, so a becomes z, b becomes y, c becomes x, d becomes…..

What is the weakness of these cypher approaches?

Frequency analysis



The Man in the Middle views all our messages and knows that letters do not get used randomly, in any sufficiently long message this histogram will reveal the link between code letters and message letters

- Side note, you should be able to apply this to wheel of fortune or hangman

Responding to Frequency analysis - Polyalphabetic Cyphers

Leon Bastilla Alberti- 1467 (ish) – use a different alphabet for different portions of text, maybe each letter.

- *An instance of* ***"security through obscurity***" – the idea that if you didn't tell people how your system worked they couldn't access your messages
- In the modern world, this is also like not telling people what the URL of your secure website is….

The specific approach was a great example of why the broader concepts was a bad idea as well!

- If the Man in the Middle knew the cypher's algorithm the code was broken not just for you but for everyone using the system!

Kerchkoffs's Principle

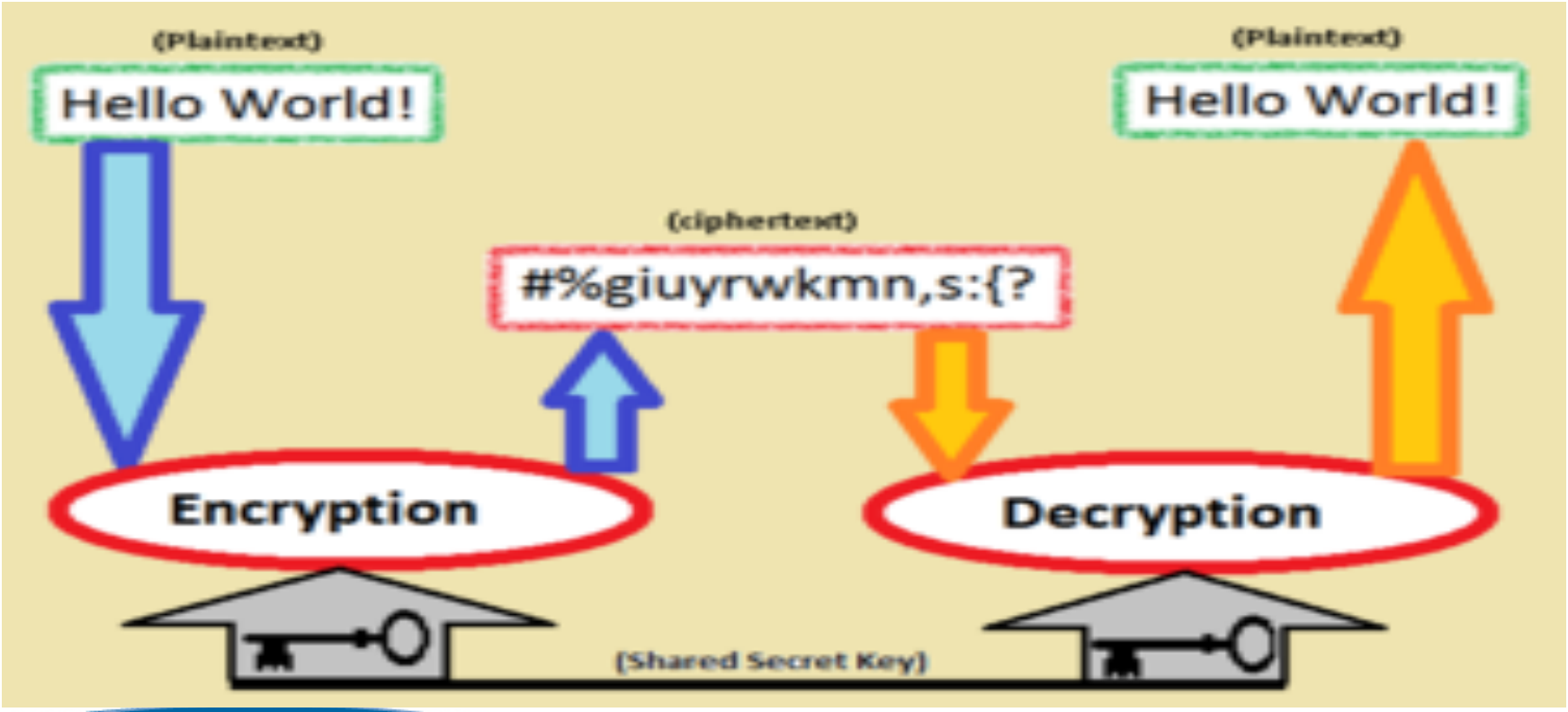
Kerchkoffs's Principle (1837) – The security of a key alone must be sufficient to guarantee the security of a message using the system

- ***A key*** – a ***shared secret*** item that unlocks something, usually a message

This was a much better approach but how do we realise this using computers?

- Typically, keys in modern computing are very large prime numbers used to alter and adjust a message in a way that appears almost totally random so removes the possibility of a frequency attack or it's more modern day equivalants

Key based encryption



Attacks in the Computer Era

The rise of the computer also made brute force attacking feasible – see Bletchley Park for the most famous example of this

- To counter this, good encryption techniques rely on easy encryption with difficult (in terms of time or maths) decryption unless you have the key
- We address this with larger key sizes, this is what people mean when they talk about 256bit or 1024bit encryption

In essence, to access this system you must require users know a *shared secret*

- But if the key is lost, stolen, or intercepted the cypher is useless – even dangerous

Kerchkoffs's Principle in Java(ish!)

```
// Code to run on Sender's machine
private Message encrypt(Message originalMessage, Key k)
{
      new Message em = SomeFunkyMaths(originalMessage,k);
      return em;
}

// em is sent over the internet to the recipient

// code to run on Recipients machine
private Message legitimateDecrypt(EncryptedMessage em, Key k)
{
      new Message decryptedMessage = reverseFunkyMaths(em,k);
      return decrypted;
}

// now decryptedMesssage should = originalMessage
```

Kerchkoffs's Principle in Java(ish!)

```
// Code to run on Sender's machine
private Message encrypt(Message originalMessage, Key k)
{
      new Message em = SomeFunkyMaths(originalMessage,k);
      return em;
}

// em is sent over the internet to the recipient

// code to run on Man in the Middle's machine
private Message ilegitimateDecrypt(EncryptedMessage em)
{
      new Message decryptedMessage = reverseFunkyMaths(em);
      return decrypted;
}

// now decryptedMesssage should = originalMessage BUT IT
TAKES 2,000 YEARS FOR THE METHOD TO RETURN
```

Final thought of the week:

What is the problem with needing a shared key to perform encryption….

Topic Learning Goals

What key assumption must we make when transferring data between two secure terminals over a network?

What vulnerability do we need to address with any encryption?

What is Kerchkoffs's principle and why is it superior to security through obscurity?

Topic Learning Goals

What key assumption must we make when transferring data between two secure terminals over a network?

There is always a man in the middle who can see, intercept and repeat our messages

What vulnerability do we need to address with any encryption?

Frequency analysis, looking for repeated elements of the encrypted message (modern equivalents are complex)

What is Kerchkoffs's principle and why is it superior to security through obscurity?

Share a secret, not a system. Compromising a shared secret compromises messages sent with that secret, compromising in obscurity compromises everyone!