Everyone download a copy of the notes before the exam. We don't want any accusations of collaboration. Good Luck :D

- File -> make a copy
- **Anyone have the power to make the file read-only by the time the exam starts?**

COMP6441/COMP6841/LAWS3040 Lecture Notes

Lecture Collaborative Notes
Wiki:
https://www.openlearning.com/unswcourses/courses/sec-21t1/week01/slides/?cl=1
Zoom: https://unsw.zoom.us/j/673508672

Clean up of document done by me! DIO!

thank you dio

Security Engineering Lecture 1 - Welcome to the Course - 15.02.2021

What course is about

Importance of Analysis

- Field is changing very quickly and in unexpected ways, but analysis is always an important part

House - How could someone break into your house?

- **Steal and copy the key**
- **Pretend to be a neighbor**
- **Create a crisis**
- **Check the door if unlocked**
- **Break the wall**
- **Through the dog door**
- **Through Garage**
- **In the dog (something about this doesn't feel right ngl)**
- **Through chimney (Santa Attack)**
- **Through windows**
- **Through Floor**
- **Through air vent**
- **Through ceiling**
- **Bruteforce key blueprint**
- **Lock picking**
- **Spare Key**
- **Tailgating**
- **Through Balcony**
- **Social Manipulation**

Story of "perfect defence house" -- Attacker will attack at weakest point, not only the front door

Finding the balance between human and machine within the system for better security

Security society UNSW (workshops, talks, CTFs and social events)
www.unswsecurity.com/join

WOW YOU HAVEN'T STARTED STUDYING YET? YOU ARE VERY BEHIND AND YOU SHOULD RETHINK YOUR LIFE CHOICES FOR A SOLID 30 MINUTES. NOW AFTER YOU DO THAT GET STARTED ON STUDYING NOW, DO YOU KNOW WHAT ECB STANDS FOR? BUT WAIT YOU HAVEN'T EVEN DONE A PAST PAPER YET. I AM VERY DISAPPOINTED IN YOU SIR/MADAM. NEXT TIME START STUDYING EARLIER. DON'T HOPE THAT OUR EXAM WOULD BE LIKE MATH1131 WHERE YOU HAVE TO DO YOUR EXAM AGAIN AND HAVE MORE TIME TO STUDY. THATS BS, START STUDYING NOW.

P.S start studying - good advice!!

Core Week 1 - Tuesday 16.02.21

History (unabridged)

- Richard talks about evolution from single celled organisms to multicellular stuff
- Then BOOM, Dinosaurs
- Then BOOM, meteor (and they were mexican (bit of a dark phase then))
- Then BOOM, chickens
- Then BOOM, the present (hackers)
- Money = oxygen
- Security started much the same - originally computers were separate and not connected to each other (single celled organisms <--> single machines)
- Once they became connected (multi celled organisms <--> computer networks) people started to hack other machines
- The modern meteorite will be explained in future lecture

Basically, then dial up existed which meant we could connect to others computers
This didn't matter much till Money was in it.
Cybercrime exceeds the money made from all the other types of crime put together

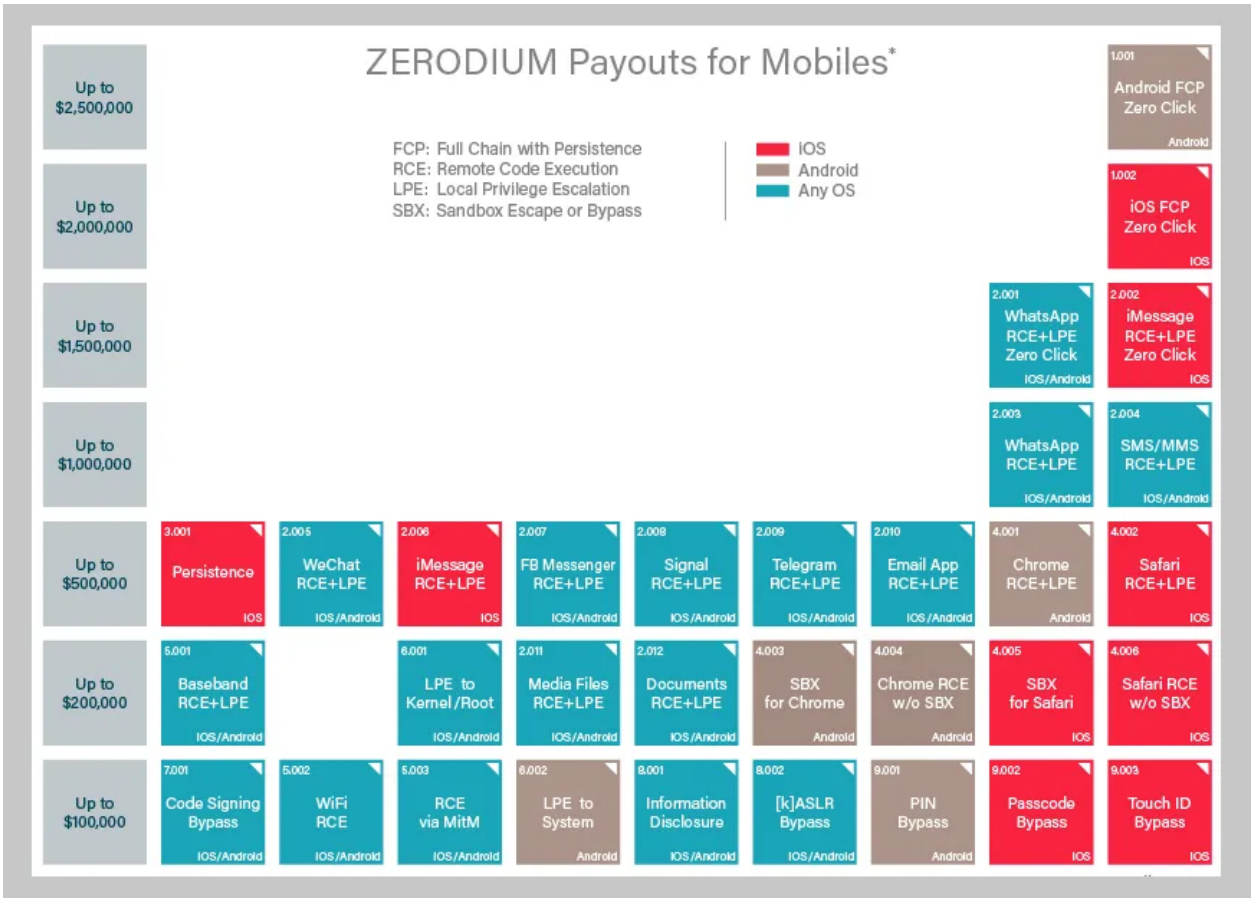
Now, exploits are made and used by lots of people working together. Not just some script kiddie.

Never Talk about:

- White hatters/Black Hatters
    - Motivation doesn't matter
    - Say Gold Hatter Instead (Pay me and I'll do it)

0Days - vulnerabilities that are not yet noticed by the "good guys"
0Days: (update this table for 2021)

ADOBE READER	$5,000–$30,000
MAC OSX	$20,000–$50,000
ANDROID	$30,000–$60,000
FLASH OR JAVA BROWSER PLUG-INS	$40,000–$100,000
MICROSOFT WORD	$50,000–$100,000
WINDOWS	$60,000–$120,000
FIREFOX OR SAFARI	$60,000–$150,000
CHROME OR INTERNET EXPLORER	$80,000–$200,000
IOS	$100,000–$250,000



NSA loves collecting these kinds of ODays so that they can access devices when they need.

Defender Mindset / Attacker Mindset / Sec-Eng Mindset

Question: Are the attackers winning?

- Somewhat.
- There's more people attacking than ever before

Whenever humans create something, we think it's super cool, and are amazed by how awesome this new creation is. But eventually we realise that there are always problems. "The problem of x."

- Computers have many "problems of x."

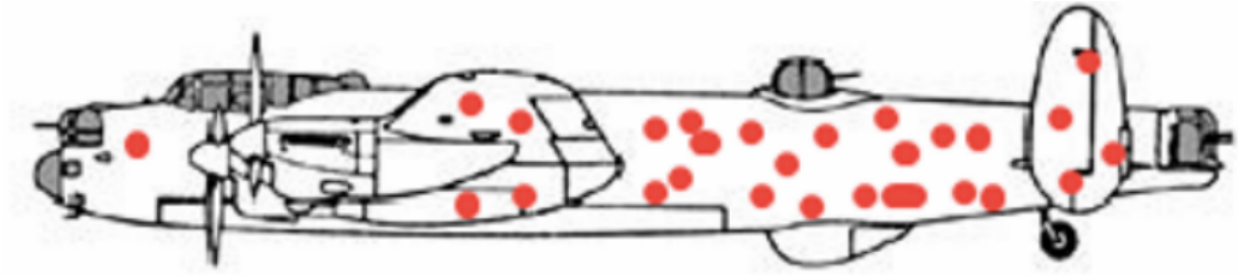
Once attackers get into a system, they can very easily stay in a system for a long time before they are detected.

Colgate: no toothpaste inside boxes, packing machine occasionally skipped boxes. Had to test the weight of pallets to find out when it's happening.

- Security mindset - have to keep looking once you solve a problem, never think you found the perfect solution.
- Found an easier solution, just have a fan beside the assembly line; empty boxes got blown away, filled ones stayed.

Mandiant report: avg number of days attackers are in network is significantly over half a year before being detected, couple of years ago they reported it was 243 days (should find a more recent number for this)

Some guy (Abraham Wald) was researching where to protect planes in the war



Where to protect:

- In front of the gun
- Landing gear
- The crew members

- The only planes that come back are the ones that were successful - where they were shot but that didn't bring them down (Survival Bias)

People often see what they want to see, so as a defender you see the defences made and feel secure. But instead, we need to think about the opposite frame of mind, much like the case with how we only saw the successful planes that had come back, and started analysing how to protect them (rather than think of the ones that were downed and actually needed protection in some unknown places.) The mind reaches out for the familiar.

Anatomy of an Attack

Depressing :(
Most attacks are going to be a form of social engineering to gain access, then jumping around to find what you need, then leaving.
i.e Humans are often the weakest link in the security chain.

Gaining access to secure locations will be a lot more difficult.

Attacks scale more easily.

Stats:
- Survival Time
- A computer can be taken over in under 60s - a shorter amount of time than how long most update downloads will take. So as soon you connect to the internet, you have vulnerabilities.

Security Patterns:
- A few tried and true, but they are always changing and being updated.
- When Data and Control is mixed, you are commonly going to get a problem.
    - E.g. telephone calls: playing audio at 2600Hz would make the call free (an internal-use frequency)

Reconnaissance:
Attackers will first research information about their target and scan for vulnerabilities in their network. There are two types of recon:

- Active recon: seeking info that can be detected or identified by the target (engaging with the target for info)
- Passive recon: Collecting info without engaging with the target

Attack:
After getting access to the network, an attacker proceeds to infiltrate the network using elevated privileges to obtain sensitive information, encrypt it or in some cases alter or erase it.

Expansion:
The next phase is to scale up their attack by expansion. They will intrude all systems connected to the network using malicious programs to enable attackers to hide in multiple systems.

Obfuscation:
Attackers will finally attempt to hide their attacks to prevent traceability or to safely place their exploit in a system without getting detected. This is mainly to confuse or disorient the defenders to slow down their response to the attack.

Engineering

What is engineering?

- Building/making things
- Embracing mistakes
- Methodical process
- Communication
- Solving right problems
- Get data/measuring
- Find the mistakes
- Mental illness

Spurious Measurement/Accuracy: Overprecision
Reasonableness Check and Approximation/estimation go hand in hand

Ants:

- How many ants would it take to fill the Ainsworth Lecture theatre
- What not to do
    - Think that's too hard, I can't do it
    - They work it out theoretically but don't actually do it, which is just trick themselves
    - Calculate volume of ant and volume of room and divide one by the other (Richard hates this because you can come up with a number and you **have no clue if it's right or not**, you just need to trust it, and he wants a **reasonableness check**)
- What to do
    - Guesstimate
    - Aim for an order of magnitude of about 2
    - GLHF, GG WP NO RE
    - **Path to security is controlling mistakes**: everything we make has mistakes in it
- **Reasonableness Check**
    1. How big is an ant, how many ants would make a green pea (say 5?)
    2. How many peas fit into some other object of slightly larger size (e.g. whiteboard eraser)
    3. How many of the whiteboard erasers fill a section of the room (something easy to estimate within reason)
    4. Repeat increasing the size of the next step up until you get to the size of the room
    - With each step, choose a size that you are confident in the estimation of the number of previous sized objects fit inside of
    -

BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP
BEEEEEEEEEEEEEEEEEEEEEEEEP

Achievement Get: Survived Fire (haha)

Recommended Reading

- Cuckoos Egg (Clifford Stoll)
- *Bill Cheswick: An Evening with Berferd. In Which a Cracker is Lured, Endured, and Studied.
- *The Art of War (Sun Tzu)

Designing against an/the adversary

Why things fail

Bingo List

- Threat Actor: Malicious person or entity responsive for incident that impact or has potential to impact, the safety and security of others
- This above is threat agent
- APT (Advanced Persistent Threat): Stealthy threat actor. Uses continuous and sophisticated hacking techniques to gain access to a system and remain inside for a prolonged and potentially destructive period of time. Their ultimate goal is to steal data, normally aim at state, national and large corporations.
- Cyber: Relating to computers
- Unhackable: Inability to be ha`ely are things "unhackable"
- Top Men: Indiana Jones reference of government "experts" who don't necessarily know what they're doing
- Blockchain: Digital record of transactions

Words to know

- Complexity - any system in which the parts of the system and their interactions together represent a specific behaviour, such that an analysis of all its constituent parts cannot explain the behaviour
    - One of the most popular examples used in this context is of an urban traffic system and emergence of traffic jams; analysis of individual

cars and car drivers cannot help explain the patterns and emergence of traffic jams

- **Coherence:**

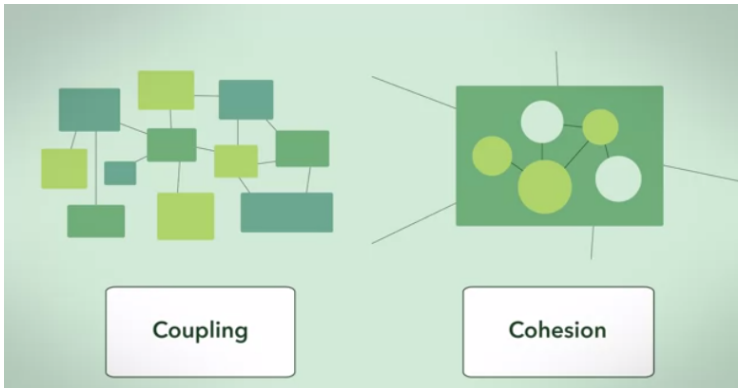
Computer science [edit]

- Coherence (programming language), an experimental programming language based upon Subtext
- Cache coherence, a special case of memory coherence
- Memory coherence, a concept in computer architecture

IT products [edit]

- Coherence (software), a component of Parallels Desktop for Mac, the Windows virtualization software
- Coherence (UPNP), some free DLNA/UPnP tools (MediaServer/MediaRender) with a Python framework
- Coherent (operating system), a UNIX-clone operating system
- Oracle Coherence, an in-memory data grid product from Oracle

-
- Coupling: Degree of interdependence between software modules



- Resilience: Ability of a solution to absorb the impact of a problem whilst continuing to provide acceptable level of service to user/business
- Single points of failure - a part of a system that, if it fails, will stop the entire system from working. Undesirable in systems that are supposed to be available and reliable.
- DOS - denial of service - an attack which seeks to make a network unavailable to the intended users by disrupting services of the host connected to the internet
- DDoS (Distributed Denial of Service): When multiple systems flood the resources of the target system

A Denial-of-Service, or DoS attack, is an attack that tries to prevent access to a service for legitimate users by overwhelming the network or server.

The Ping of Death or POD, is a pretty simple example of a DoS attack. It works by sending a malformed ping to a computer. The ping would be larger in size than what the internet protocol was made to handle. So it results in a buffer overflow.
Another example is a ping flood, which sends tons of ping packets to a system. More specifically, it sends ICMP echo requests, since a ping expects an equal number of ICMP echo replies. If a computer can't keep up with this, then it's prone to being overwhelmed and taken down.
In a SYN flood, the server is being bombarded with the SYN packets. The server is sending back SYN-ACK packets but the attacker is not sending ack messages. This means that the connection stays open and is taking up the server's resources. refer to SYN floods as half-open attacks
A DoS attack using multiple systems, is called a distributed denial-of-service attack or DDoS. DDoS attacks need a large volume of systems to carry out an attack and they're usually helped by botnet attackers. In that scenario, they can gain access to large volumes of machines to perform an attack.

- Tamper Proof: Made so that it cannot be interfered with
- Tamper Evident: Packaging/system designed to clearly reveal any interference with its contents
- Supply Chain: sequence of processes involved in the production and distribution of a product/service/produce
- Malware: Software that is specifically designed to disrupt, damage, or gain unauthorized access to a computer system
- Ransomware: Malicious software designed to block access to a computer system until a sum of money is paid

-

Think of the big picture for case studies. Even if you think you got it, keep thinking.

Week 2 - Monday 22/02/2021:

Richard's friend has a new puppy. That's it for now. Not even pet tax.

Trust/Abuse:

Every (cyber) attack is an abuse of trust.

Risk game, Chess Examples where there is an implicit trust which is required for the functioning of the system (you don't suddenly expect a pawn to kill your own pieces, or refuse to promote into a Queen).

Ability to work together and collaborate is our super power

Particularly insidious aspect of society in general and security specifically as well

Richard and his brother liked their mum's cheesecake:
Can Richard's brother trust him to cut an equal slice of cheesecake? **NO**

What if their mum (trusted third party) cut the slice instead? But that's so much work for his mum :(
Strategy to allow equal slices without third party -> one party cuts and other party chooses

- "You cut, I choose"

How do we extend this to multiple parties ?

This situation is analogous to the scene in War Games where they have to confirm whether or not the missiles are actually coming - However, how would you tell, without a third party, if the missiles are actually coming ?

Your computer CANNOT actually tell if it is you !! It just be reading bits

Are we in a simulation?

Slightly related is: the two generals problem (model for a concurrency problem!)

Richard pretending to be a pigeon
Point is that you cannot have a guaranteed Protocol which allows them both to be 100% sure whether or not they should attack. 'Impossibility of communication'

So it becomes a suspicion chain

Opposite of Ronald Reagan: Do things right rather than get everything in -> THIS APPLIES TO ACTIVITIES

Type I and Type II Error (This is a common theme in Security problems)

Say you are testing for Covid:

Test say yes and you have it (true positive)	Test say no but you do have it (false negative = type II)
Test say yes but you don't have it (false positive = type I)	Test say no and you don't have it (true negative)

zero
Say we tweak our test (to be more sensitive or less sensitive to the virus), but improving a Type I or Type II error is a Zero Sum game: one will improve, another will suffer (this tradeoff is well-explained in the Welfare example a few paragraphs later).

[The following is formal Engineering Definition of Type I & II, which Richard thankfully does not use]
A type I error is the rejection of a true null hypothesis (also known as a "false positive" finding or conclusion; example: "an innocent person is convicted"), while a type II error is

the non-rejection of a false null hypothesis (also known as a "false negative" finding or conclusion; example: "a guilty person is not convicted").

Note that which is type 1 error and which is type 2 depends upon initial assumptions of the problem- the table above implicitly assumes everyone does NOT have COVID. If we instead assume that everyone does have COVID, the error types switch around.

** I feel like row1 col1 and row2 col2 should swap if it's ppl NOT have COVID case.**

I.e. Improving welfare

Type I issue: The most disadvantaged cannot access welfare payments

Type II issue: Some are taking advantage of the welfare payments when they don't need it.

Type I solution: Make welfare more accessible (Disadvantage: Type II suffers; more ineligible people access welfare)

Type II solution: Make welfare less accessible (Disadvantage: Type I suffers; more eligible people will not gain access welfare)

Fixing a security error may decrease the chance of that error occurring, however, it may increase chances of another error as a by-product.

- Improvements will usually err on the side of the **visible problem**
- Improving on the **invisible problem** can cause annoyance

LEARNING FORCE UNITE

Probability

Chances are the long run probabilities of a situation occurring
Lion coming in -> High impact but low likelihood

Risk: Impact + Probability

e.g. Smoking a durry (aussie slang for cigs): Low Impact + High Probability

e.g. Living next to an active volcano: High Impact + Low Probability

Humans are terrible at dealing with High Impact + Low Probability problems. Humans don't have much experience dealing with these problems. E.g. people that continue to live in areas that lie on fault lines (San Andreas), or next to active volcanoes (e.g. Naples, Italy).

is difficult to persuade investment into High Impact + Low Probability problems.

Engineering solution: Spread our attention/resources across all problems.

High likelihood but low impact risks are good because we know how to deal with them because they happen often
The one we have to worry about is low likelihood high impact

Don't be a Monday's Expert (Hindsight Bias / Creeping Determinism)

Be careful of about hidden dependencies, in that other risks are more likely to occur, if one risk occurred

Independent vs Correlated Risks:
Easier to assume things are independent because the maths is easier

Ice-cream causes violence in summer (causation vs correlation)

Human error tends to cancel each other out however computer errors are systematic and are more dangerous and tend to repeat. ie counting votes

Dealing with Risk:

- Mitigate the risk: Reduce the impact, does not reduce the probability.
- Pass on the risk: e.g. casinos pass risk to government (invisible), privatise the profits, nationalise the risk
- Immunise: Invest in the risk actually occurring (balancing out)- hedging
    - e.g. an airline company risks oil prices rising, they will then invest in oil to balance out their portfolio in case the risk comes to fruition.
- Accept: ehh its ok
- Pool: share risk amongst lots of people

Think about this: How do we assess risk? What information is important to have when analysing risk ?

Week 2 - Tuesday 23.02.2021

Week 2 Yeet
Talked about engineering type solutions last week

Navid got a cameo on the big screen
BSides Tickets available (BSides Canberra) For April 9th - 10th

Audio go *WEWEWEWEWEWEWEWEWEWEWEW*

Physical Security

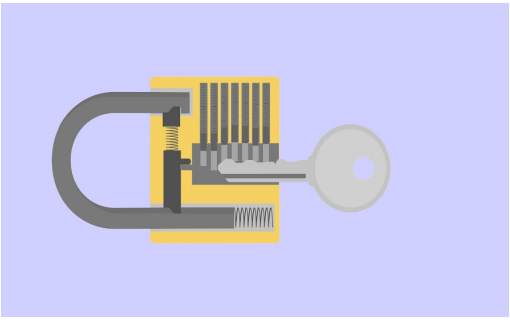
- Locks on doors
- Hiding things away
- Security Cameras
- All the non-computing stuff

If a person can get a hold of your device, it's game over. Hence your physical security needs to be up to scratch to protect everything else.

Really hard to achieve physical security. Look into locks/lockpicking. Locks are just flimflammery, Aren't actually super secure

Examples of other things which are not secure:

1. Burning of paper doesn't work either
2. Writing on a piece of paper leaves a trace on the paper below
3. Different clicks on a keyboard are different sounds



To keep something safe, you have to have to have a *plan to keep it safe*.
Look at destroying data, really hard to destroy

“Every contact leaves a trace”
Everything you do will leave a trace.
Side channel attacks

The computing world relies on the physical world. Hence physical security is very important. If destroyed in the physical world, it is game over.
Ie, The Matrix

Tampering: Tamper evident vs tamper proof

- Tamper Evident : Shows tamper if it has been edited (Example: a jar of jam because the cap pops up)
- Tamper Proof : Prevents tamper

Never forget about the physical world.

Recon

The act of looking into something/someone before you attack them.
All sophisticated attacks will do this.

Passive vs Active Recon

Passive Recon	Active recon
Just searching online	Talking to people, searching for people, entering location

Activity: How much info can you get about a company or a person with just research

Richard provides an example of how one could : Get a home address from background of a photo (using pen and papers)

1. Get date from photo
2. Use angle of sun in the photo to determine which street side

3. Get an idea of the current suburb by searching up on this famous person's wife and kids school etc.
4. Use google street view to walk through the suburb and get to the house

Tldr. finding info about people in publicly available places is extremely easy

OSINT - Open Source Intelligence
The practise of collecting Information that is openly available.

- Meta-data from photos
- Online Papers
- Etc.

Bellingcat: International Organization, that search for information using only OSINT

Shredding exercise!

Secrets (And why is it so hard to keep them)

Who do you trust?

- Don't tell people secrets
- IF you want to keep a secret THEN tell nobody
- IF you tell someone a secret THEN you can't tell if they tell anyone else that secret

How can you keep a secret?

- There is no way of keeping anything secret

What do you do if your security relies on a secret? Well...All of cyber security relies on secrets…

Ruh roh raggy

A bare minimum of properties we need to do anything on the internet:

1. **Confidentiality** - How we can pass information around without others finding out about it.
2. **Integrity** - the quality of being honest. (If I write you a check, integrity implies that the check has not been forged with)
3. **Authentication** - the process or action of verifying the identity of a user

*Richards CIA

Good book -> The Code Breakers by David Kahn

The history of keeping secrets

Risk - Secrets will spread out once they are exposed to someone else.

Sending Messages - How do I spread messages when I can't just tell it to people.
But what if someone messes with it?

Method 1 : Caesar Cipher (advancing by a fixed number of letters)

How can I create asymmetry (i.e. a property the only recipient knows and not anyone else)

Shared Secrets. Everyone can see the message but you need the secret to decipher them.
Problems with this: Secrets can be shared and secrets leak.

Caesar's method: shift all messages by 3 letters (someone said 13 in class but is actually 3) 13 is ROT13.
This is called a **Substitution Cipher**

z

Method 2: Codes

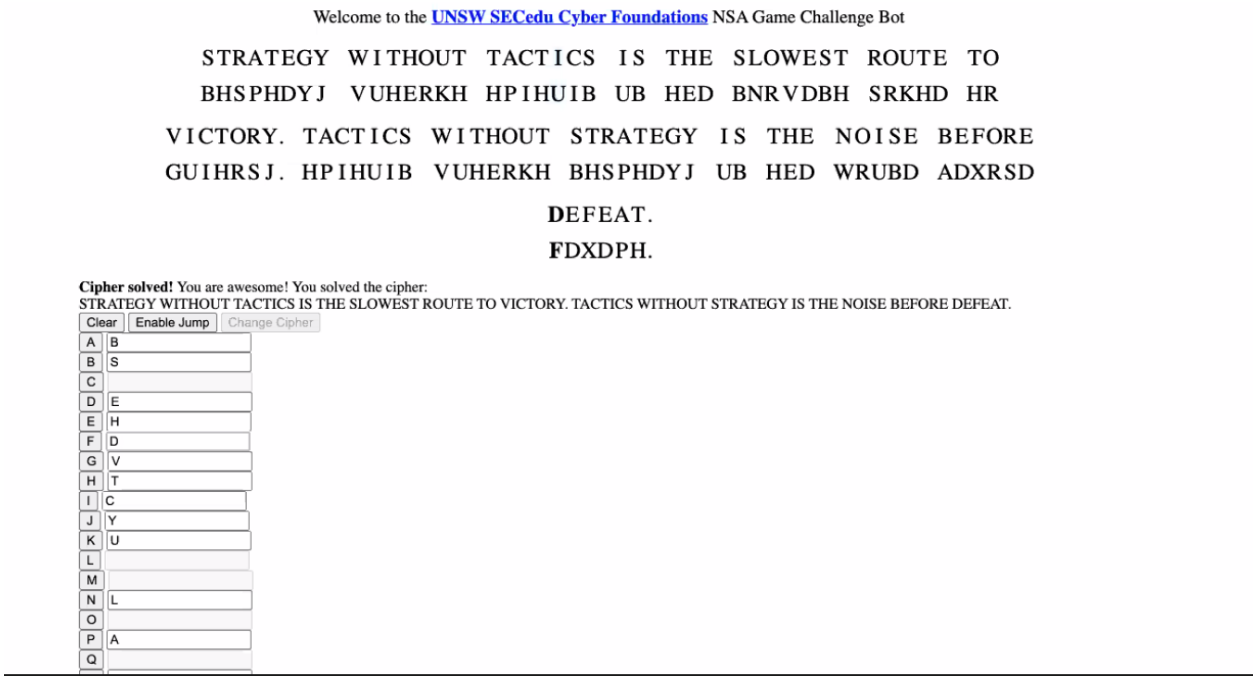
Instead of individual letters, replace whole words.
Problem with codes:

- Cumbersome, keeps getting adding to
- Have to maintain codebook (list of correspondences between codewords and real words)
- Not easily repaired if something breaks (method is the secret)

Method 3: Substitution Cipher
Method 4: Vigenere Cipher
Method 5: Permutation codes

Andrew then shredded paper and said there was a bonus mark for reassembling it

Decoding exercise - UNSW SECedu cyber foundation nsa game challenge bot



ETAOIN SHRDLU -to remember most frequent letters
All tricks are breakable

Steganography: hiding the existence of the message.Invisible ink

- Romans writing messages on the slaves head
- Newspaper prick method

Problems:

- Once the method is known, the whole secret is out.

Workaround: let's not make the mechanism a secret. Assume the enemy should know everything except the key.

The above method is pretty much the only method that will work because all the others rely on "security through obscurity" - i.e. what you think is obscure is not obscure. The enemy knows everything except for the key.

Tldr. Security should (not) be relying on a secret. Because nothing can be a secret
Richard gave an example of an awesome project before - sydney trains tickets coding. They finally broke the ticket machine.

Attacker's pov:

- Can take advantage of asymmetry: they have to attack only one spot
- Pick the weakest spots

Defender's pov:

- Has to defend every spot to win
- Make every spot strong
- Defender's main advantage: community & collaboration

Enemy should know the system

Week 3 - Monday 01/03/2021

Decode Puzzle

Opening class with a puzzle, the Professor called a student (Jing) to predict a random sentence from a novel, letter by letter.

During each letter, the student was being told whether the letter he got was correct or not, and if incorrect, then what the correct letter was. After “I tr”, he could guess that the next would be i, then e, then d (“tried”), and the same happened multiple times, when he could guess the next letter from the previous context.

Final Sentence: “I tried to not respond immediately, but I failed.”

The student got 28 out of 39 letters correct.

Professor claimed that given the input of wrong guesses, the student can try again and get the whole sentence again. So only 39-28 = 11 letters are necessary, the rest are redundant, as he could predict them. So we can compress this sentence.

11 letters mean 11 places, each with 26 options, so 26^11 total possibilities (being case-insensitive, so just taking 26 letters of the English alphabet).
= 3.6 quadrillion possibilities. Order of 10^15.
(Used Wolfram Alpha for calculation)

Other students argued that given the wrong letter as input, the volunteer can still decode the sentence wrong as there are still a lot of possibilities. However, Richard said that Jin will be able to decode the full sentence via his brain meat algorithm.

“Wow, Jing is so cool, I wish I could be cool like Jing" - totally not Jing

Binary Numbers and Numerical Representation

Two aspects to numbers

- mathematical definition (always consistent). E.g. 7 is 4+3, and 5+2, and 1+1… 7 times, irrespective of cultural context

- linguistic representation (not always consistent). E.g. different languages have different words for “7”.

Binary numbers are just another way to represent the same concept

Place value system comes from Hindu-Arabic numerals. More efficient than building a number by adding 1’s, as adds powers of 10:

42 is 4 10’s and 2 1’s (4x10 + 2x1)

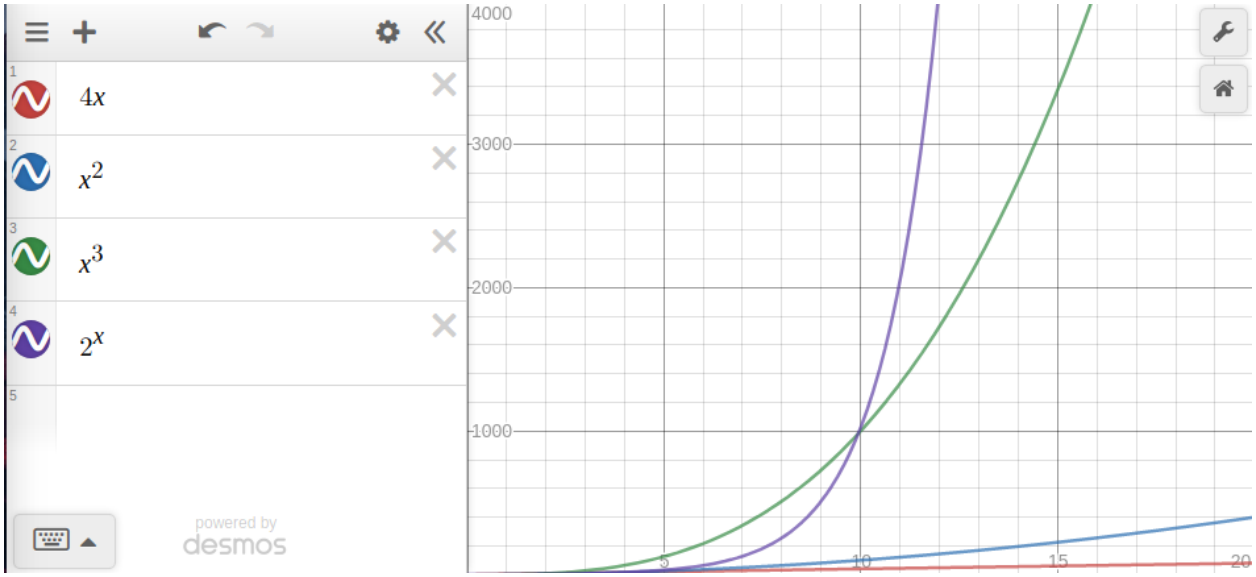
123 is 1 100’s, 2 10’s, and 3 1’s.

Decimal number	Binary number
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

Exponentiation

- ‘Raising to the power’
- Base to the power of exponent = result
- Rate of increase increases as you continue

Comparing growth rates



Slow growth - 4x or 5x etc (multiplying by a number. Called 'Linear', coz graph is a line.)
Medium growth - x^2 or x^3 etc (variable as base. Graph is a curve.)
High growth - 2^x (variable as exponent. Graph is a fast-rising curve. Initially slower than above, then much faster)

Can see in graph above that 2^x (purple line) is lower than green line initially, but then rises much faster.

-/riddle:

Martian Men lie
Martian women tell Truth
Venetian men tell truth

Venetian women lie.

(The book 'The Chess Mysteries of Sherlock Holmes' by this author was also recommended.)

What yes/no question would tell you if someone is from Venus:
"Are you a man?"
Only Venetians would answer yes to the above. (ref table below for exhaustive options)

Question to ask to check if someone is a woman:
"Are you from Mars?"
Only women would answer yes to the above.

Impossible to distinguish four sets with a single question of yes or no. Need to ask at least two questions to determine both planet and gender.
Information theory (Claude Shannon was a pioneer of it).

Question	**Martian Men**	**Martian Women**	**Venus Men**	**Venus Women**
"Are you a woman?"	Yes	Yes	No	No
"Are you a man?"	No	No	Yes	Yesti
"Are you from Mars?"	No	Yes	No	Yes

$10^3 = 1000$ is approx equal to 2^{10} (1024). This can be used to rapidly estimate how many bits are needed to store some information. Won't be exactly right at high powers, but good enough to get a ballpark figure (refers to first lecture: engineers estimate to a 'good-enough' level, don't waste resources in trying to get to an exact figure, if it's not needed.)
E.g. the 26^{11} we calculated above as approx 10^{15} can be estimated in bits as : $10^{15} = (10^3)^5 = \sim (2^{10})^5 = 2^{50}$. So 50 bits.

The game of 20 questions, each with a yes/no answer, lets you span 20 bits of information, so 2^20 = (2^10)^2 =~ (1000)^2 = 1M possibilities. (Example of actual 20 questions game played in class a little later in this doc)

How to measure things. Is a pic worth 1000 words? (Implies that with well-used 1000 words, you can convey more than a picture. Good example in the game that is coming up)

Compression

Lossless compression

- Able to be decompressed to the original
- Only around 1.75 bits per English character vs several bits per char in most text encoding hence lossless compression usually effective (e.g. ASCII encoding uses 8-bits per character).

20 questions with Zoom students

Richard is thinking of something for the online students at Zoom to guess.
Want questions that are not too specific.t

Q1: Is it an object? YES
Q2: Is it a person? NO
Q3: Is it something you use? (Hard for Richard) NO
Q4: Is it heavy? YES
Q5: Is it smaller than a bread box? NO
Q6: Is it furniture? NO
Q7: Is it manmade? YES
Q8: Is it smaller than a car? NO
Q9: Does it work on electricity? TOO COMPLICATED TO ANSWER
Q10: Do I own these objects? NO
Q11: Do I like it? YES
Q12: Do people go inside it? YES
Q13: Does it have bright colour? NO

Q14: Is it something you can see? YES
Q15: Does the name of the object begin with the letterx …

RICHARD SPILLED IT - IT'S THE SYDNEY OPERA HOUSE!!!

In 20 questions, you can pretty much distinguish every object from another one.

Every additional question doubles the Universe we can address, massively increases the things we can describe. This is the power of exponentiation. 2^1, then 2^2, …. 2^20.
2^20 = (2^10)^2 = ~ (10^3)^2 = 1,000,000 , i.e. a million things.

In such games, it's much better to ask questions like "Is it smaller than a bread box?", than "Is it a pencil?". Every question partitions the Universe into two. E.g. Pencil question partitions the universe of possibilities into pencils vs all objects that are not a pencil. It's better to ask questions that partition it more efficiently, into approximately equal halves, like "bigger than bread box" and "smaller than bread box". If we ask a very specific question, like the pencil question, and the object is not a pencil, then it could be almost anything else, and we haven't gained much information. This is more important in the early stages, when we need to rapidly partition till we reach a small set of possibilities. The last few questions can be more specific.
Usually by 12, 13 questions, people have a good idea of what category the thing is, and by 17-18, they are sure of what it is. Because whatever most people think of (and guess for) would lie in the same set of 1 million objects.

DISCUSSION Question before the break: IS A PICTURE WORTH A 1000 WORDS?

(it probably isn't, we have seen how much information can be conveyed in a thousand words)

Mastermind Game

Question - How much information can you get from each position?
What's the minimum amount of rows you need to get the right answer?

We can measure security as the amount of work needed to break it.
HOW MUCH WORK DOES A COMPUTER REQUIRE TO DO A PARTICULAR TASK?

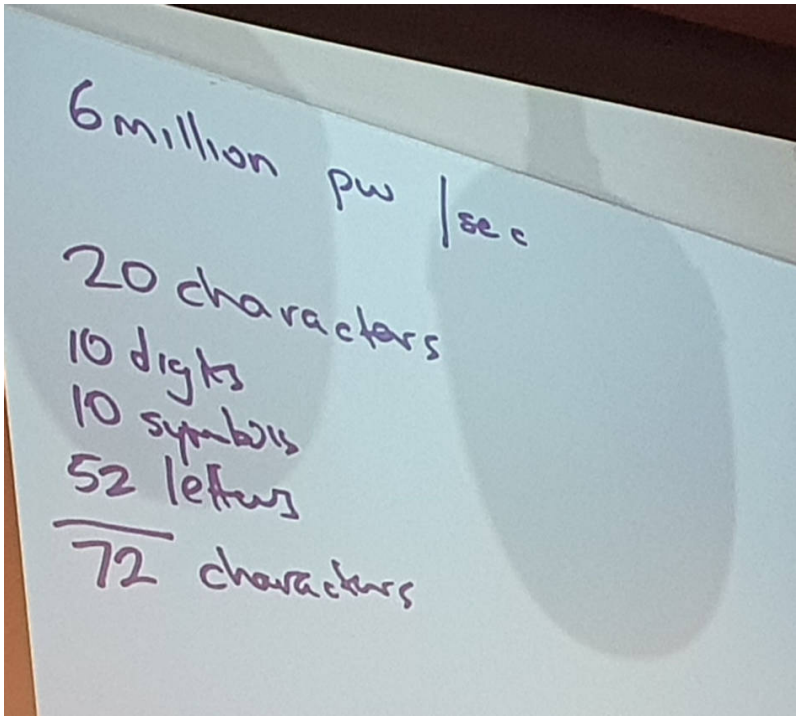
HOW LONG WILL IT TAKE FOR BRUTE FORCE TO WORK?

Measure it and determine if it's feasible if the hacker can do all of those work to achieve that task.

- Example #1: out of the 1000 doors the attacker get 300 doors but still had to attempt to force the 1000 doors
- Example #2: It takes a computer a 1000 operations to try 1 AES key. A 3 GHz computer with dual cores can do 6 billion operations a second. So it can try 6 million keys a second.
    - Computer's Hz x number of cores

Security is like a knot - it can be undone given enough effort. **Security can only buy time**.

Q: How long does it take for a computer to brute force a 20-characters password, assuming the computer can do 6 million passwords per second?



We have 20 slots, each of which can take any of 72 characters. So the value we need is 72 x 72 x 72…. 20 times = 72^20. Per Wolfram Alpha, that's approx 10^37 possible words (passwords).
Using the earlier approximation, it comes to 2^120, we add back the 10 we threw out in the mantissa, that adds 3 bits, ~2^123, so 123 bits. (10 ~= 8, so 2^3)

6 million is approx 2^23.

So 2^123/2^23 = 2^100 = ~10^30 seconds needed to try all passwords
On average, a password would be found in the middle of the set (won't need to go through all possible passwords). So I will need one bit less (**removing a bit halves the set**, goes from full set to middle of the set).

After converting seconds to years, the attacker will need 10^22 years to crack a password. Richard: "That's more than 22 years!".

But the above depends on the password being strong, not a dictionary word. Assuming a dictionary has ~30k words, then the password would be cracked in less than a second (since our attacking computer can step through 6 million words a second).

Even with strong passwords, it's only a matter of time, and compute power. With a supercomputer it becomes easier. If quantum computing works as predicted, even strong passwords would be crackable quite easily.

Moore's law. If we assume computing power doubles every year (actual law is more conservative, around 18 months), then we shave off a bit of complexity every year, from the attacker's perspective. So should the key be 10 bits, 20 bits, 100 bits, 1000 bits?

Questions to ask:
How long does it need to be secret for?
Who are we defending against?

Defending against casual home-based attacker is much easier than defending against a foreign govt or NSA.

The attacker might not brute force it. Often weaknesses are found in the algorithm, some mathematical flaw that makes it easier. Analogous to the Rubik's Cube video last week, where something scrambled in 10 steps could be unscrambled in much fewer steps.

10^22 is so long that it isn't in danger from Moore's law (the time is longer than the age of the Universe), but Moore's Law IS a danger if we are looking at time horizons like 100 years.

Also, passwords are usually not random. They are meaningful words, and easy to guess. We may think they are hard to guess, but if one human thought of it, so would another. Same as hiding a key. Most people think of the same spots. The only random way is to close your eyes, spin round and round, and then open your eyes, and then hide the key there, wherever you are pointing now. If you "think" of a spot, so would another person.

List of Supercomputers (https://www.top500.org/lists/top500/2020/11/). Fastest currently, Supercomputer Fugaku, has 7 million cores, and can do 400,000 Teraflops per second.

A person can build a custom computer just to break passwords, would be more efficient than using a general purpose computer. GPUs and FPGAs are often better for intensive calculations (e.g. mining Bitcoin).

The Matrix opening scene: the computer is cracking the digits of the phone number, but it works at constant time. But as each digit is cracked, the search space goes down by a factor of 10. So later digits should be cracked faster. But it happens at a constant speed. Why? Because Hollywood doesn't understand exponentiation.

Applied Cryptography book, Bruce Schneier

History of DES (Digital Encryption Standard). NSA 20 years ahead of everyone. The modern standard is AES.

Good example to understand exponentiation: you have a lily pond and the number of lilies are doubling every second. After exactly a minute, the pond is full. At which point was it half full?

Correct answer is 59 seconds. Coz it's doubling, so went from half to full in the last second.

Week 3 - Tuesday 02.03.2021

Understanding People?

Security - **properties** we want to establish e.g I don't want my home to spy on me, I don't want my confidential data to be shared > not mathematical
System is not just a computer, there is an environmental component
E.g hacking people by tricking the owner
What we want in security is end-to-end security (people are at each end), the whole journey needs to be secure; **Not all of security is in computing, can attack hardware or people in real life in physical world**

E.g vaccine that needs to be kept at a certain temperature until injected (needs to be like this for the entire journey from manufacture to injection)
"People are the weakest link", not entirely true or false
Know both technique and humanity
"Bitsquatting" - exploiting vulnerabilities in physical hardware of the chip

Cognitive Vulnerability

First hierarchy is mistakes : **all code have mistakes, there are vulnerabilities that people can use to make exploits**
Cognitive vulnerability: we can't rely on our brain 100% of the time
Example: hand in hot and cold water for 30s then into room temp > different sensations, optical illusions
Dealing with risks :
For familiar risks, we can decide what to tune out and focus on
For unfamiliar risk, we can focus on it obsessively or ignore it completely
Find evidence your brain has flaws, people who exploit those are called "social engineers"
Cognitive Bias (20 of them)
https://www.businessinsider.com.au/cognitive-biases-that-affect-decisions-2015-8?r=US&IR=T

Relation to cybersec or security, announcements near celebrations or crisis, timed attacks when everyone is distracted (the focus is on something else)

Social engineering

The art of being systematic about tricking people

- Art of Deception book by Mitnick

Distraction attack, Disguising as authority, People don't like being rude, People like to conform

Recon is the first step
Phishing: people tend to ignore risk when they see profit (Greed)
Chaser Trojan Horse - https://www.youtube.com/watch?v=JjShDCqs7jk

Hubris - thinking as a defender, impairs rational choice
Abuse of trust
- Phishing,
On cybertraining, how would you change people's behaviours such that they would be more secure

Social engineering is an attack method that relies heavily on interactions with humans instead of computers.

Social engineering is a kind of con game where attackers use deceptive techniques to gain access to personal information. They then try to have a user execute something, and basically scam a victim into doing that thing.

A popular type of social engineering attack is a phishing attack. Phishing usually occurs when a malicious email is sent to a victim disguised as something legitimate.

Another variation of phishing is spear phishing. Both phishing schemes have the same end goals, but spearfishing specifically targets individual or group. The fake emails may contain some personal information like your name, or the names of friends or family. So they seem more trustworthy.

Another popular social engineering attack is email spoofing. Spoofing is when a source is masquerading around as something else.

one attack happens through actual physical contact. This is called baiting, which is used to entice a victim to do something.

Another popular attack that can occur offline is called tailgating, which is essentially gaining access into a restricted area or building by following a real employee in.

Weakness of the Week

Possible weaknesses:

- Respect for authority
- Trusting
- Pride
- Kindness
- NIST 800-63 2019 password guidelines
- Ignorance
- Slothbug
- Routine Complacency
- Peer Pressure
- Habit
- Routine
- Stubbornness
- 7 Deadly Sins
- Poor Prioritising

Richard's book recommendations:

- The psychology of persuasion
- Predictably Irrational - Dan Ariely
- Blindspot

This week's weakness(es) : Gullibility and Greed

Gullibility

- Preconceived notions > we are "programmed" to comply and acquiesce with the other person.
- Eagerness to believe or trust > we ignore our "programming" or intuition and fail to assess the situation rationally; we ignore the fact that in such situations we are required to find out more information.
- We end up getting conned because of our inability to decline, refuse or postpone our decision on what was manipulatively put in front of us.

Greed

- People get greedy and cheat, ironically a person responsible for ensuring no cheating or financial matters end up cheating
- Systems should be in place to be accounted for greedy people
- People have their own interests as well which can influence the company's interest
- Decisions should be set up such that there's no extra pressure on them making the wrong decision
    - Example: Judges should not be part of a trial they have a connection to
    - Examples
    - Enron
    - 1995 Barings Bank - Nick Leeson
    - Martha Stewart - Stock Scandal
    - LIBOR scandal 2012 http://www.informath.org/media/a72/b3.html

Greedy people

- A little more interested in money
- Suspicious of other people

Password problems

Throwback to the last lecture 20 long characters take way too long to crack but are also hard to remember so we use dictionary words and alter them to make it easier to remember > this comes with a downside of smaller option space.
Humans are bad at picking passwords.
Obscurity - taking a hash of the password and sending it to the computer which stores it. This alleviates the security risk of an inside person having access to the password (incase they get reused elsewhere) good hashes can't go back > asymmetric
Attackers hack hashed passwords by looking at a list of most common passwords and then comparing it with a list of hashed passwords to figure out what the hashed passwords or even the function function
Problems with reusing password
How to generate a good password

Week 4 - Monday 8th March 2021

We typically assume that when dealing with a security problem everything is clean. In reality this is far from the truth. E.g. memory is stored from the previous user.

AES designed to use different key sizes.

Key is considered broken if it can be solved in a quicker time than brute forcing.
AES and DES are symmetric algorithms. That is they use the same key for encryption and decryption.

Sole confidence in AES is that no one has yet to crack it.

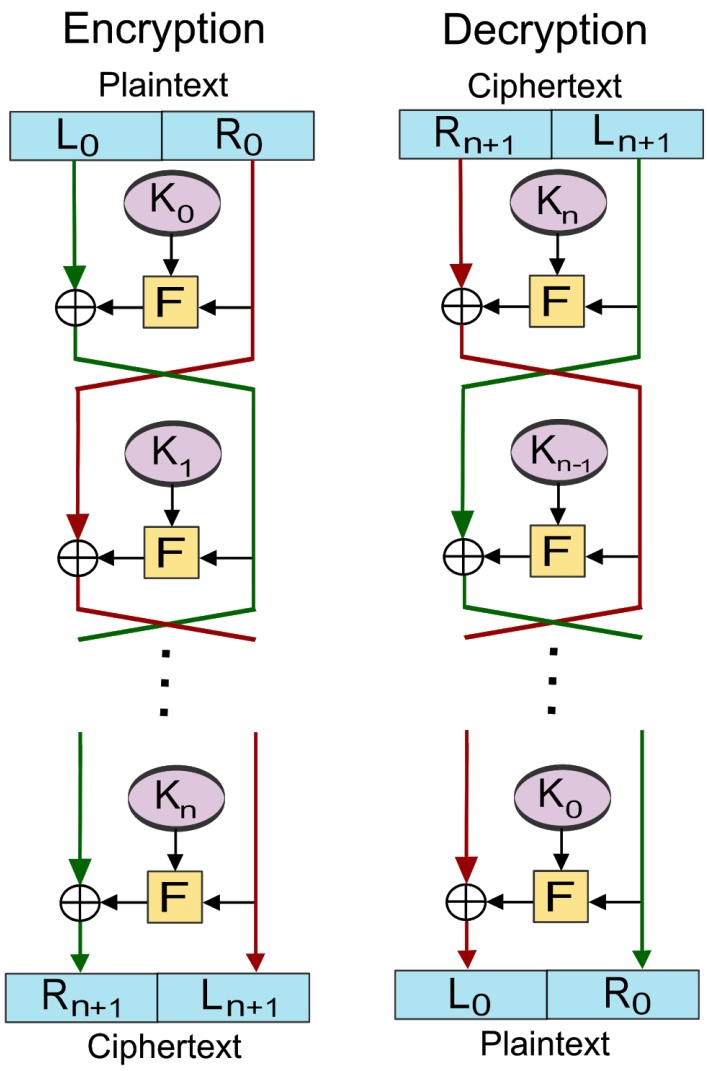
Block cipher

- Take message
- Split into chunks - equal-sized chunks, with last chunk padded if necessary
- Chunks are encrypted
- Stick the encrypted chunks together (block nodes)

Stream/string cipher

- Encrypt message one letter at a time
- Very convenient as it can encrypt as it goes along rather than waiting for chunks
- Downside of stream/string cipher is that the plaintext and ciphertext are in the same order

Feistel networks - flow chart of encryption and decryption



1. Split plaintext into L0 and R0 blocks

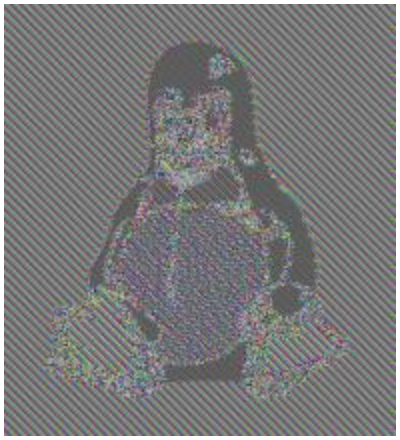
2. R0 goes into the function F (could be anything - double number + 6, reverse all bits, etc. Function must be deterministic) and combine it with key K0 to get output
3. Output (step 2) XOR L0 (XOR has nice property in that it is reversible - doing the action twice gets back to the original). This produces a jumbled L0
4. The RHS and LHS now swaps over (every time they do, it's the end of a round). Output (step 3) goes into the same function (step 2) with another key K1
5. Output (step 4) XOR R0
6. Repeat steps
7. What we have built is DES

Same logic can be applied to decryption.

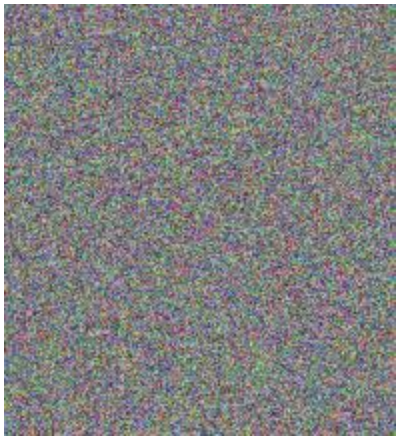
AES uses a slightly different method - SP (substitution permutation) network

How do you build the blocks/chunks together? ECB mode (**electronic code book mode**) - split, encrypt and join together in the same order. This is subject to frequency analysis, which is when attackers get statishacked as we can cross reference chunks of ciphertext to plaintext. This mode should <u>never</u> be used.

UNENCRYPTED DATA



ECB MODE – ENCRYPTED DATA



CBC MODE – ENCRYPTED DATA

Images Courtesy Wikipedia.com

Symmetric cipher OTP - one time pad is Shannon-wise a perfect crypto system and can't be cracked. You know if you are decrypting something you are right if the output makes sense because the amount of information in the key is far smaller than the information in

the message. In OTP every letter is encrypted with a different key. Key is as long as the message which means that all possible messages are possible. Therefore you need to have the key to know you have cracked the message. The downside is the key is long.

Practical shortcomings with OTP:

- Can't reuse the key numbers
- Key numbers have to be random - Russians found it hard to generate many random numbers
- Symmetric cipher - you have to share the secret. Distributing the secret is hard
- If the messages is not sent perfectly (missed packets over network) then the message is out of sync and can't be decrypted properly

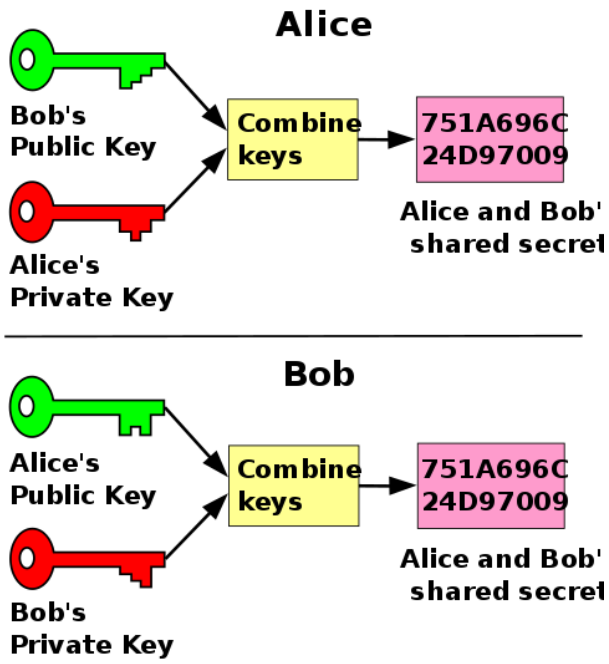
The key problem

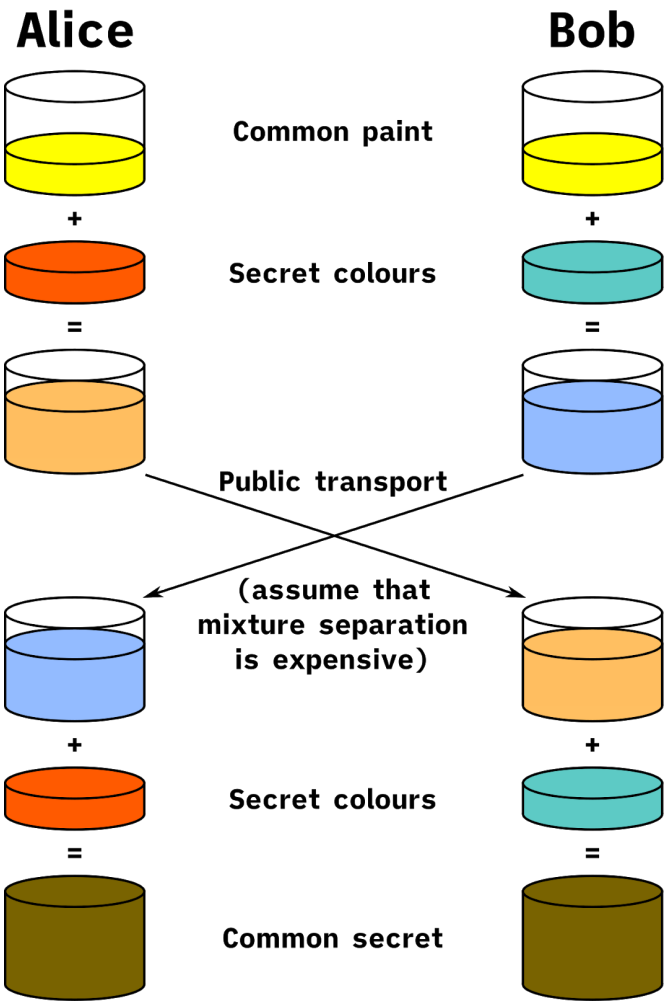
- Sharing - how to get the key to the person on the other side
- Managing
- Don't roll your own

Solutions to the key problem

- Merkle 74
    - Problem: how can I get a key to you?
    - Solution: suppose a sack of index cards and encrypt a message of each card, something that can be cracked in an hour, the plaintext is ''The 512th key is sausage p' and each message has the key for each message.You open the sack and pick a encrypted message at random and decrypt it and send a message back saying you are using the 512th password and then find the 512th key and then you can communicate using this channel.
    - Problem - It's confidential, but can't authenticate the user on the other side
- Diffie Hellman - practically didn't take off even though the idea was brilliant
    - Hard to invert the colour even if you have the common colour
    - Instead of colours we use numbers and instead of mixing we use invertible functions
    - Process
        - Takes a number we both know (common number)
        - Number is raised by a power (our keys) then modulo it
        - Send reminder to other person
        - Raise other persons to your key

        - These numbers will be the same
    - Pros:
        - Https uses Diffie Hellman - each session uses a new key each time. Produces **perfect forward security** - even if at a later date, they can't decrypt our messages as session keys are different from the master keys. Perhaps the master key was used to establish the session key, but the master key will be disposed just like the session keys after every session.
        - Can set up a key really quickly
    - Weakness - used the prime example. Paper explains that many browsers use this one number.





Example of Diffie Hellman Key Exchange taken from Wikipedia:

Here is an example of the protocol, with non-secret values in blue, and secret values in **red**.

1. Alice and Bob publicly agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).
2. Alice chooses a secret integer $\mathbf{a} = 4$, then sends Bob $A = g^{a} \bmod p$
   - $A = 5^{4} \bmod 23 = 4$
3. Bob chooses a secret integer $\mathbf{b} = 3$, then sends Alice $B = g^{b} \bmod p$
   - $B = 5^{3} \bmod 23 = 10$
4. Alice computes $\mathbf{s} = B^{a} \bmod p$
   - $\mathbf{s} = 10^{4} \bmod 23 = 18$
5. Bob computes $\mathbf{s} = A^{b} \bmod p$
   - $\mathbf{s} = 4^{3} \bmod 23 = 18$
6. Alice and Bob now share a secret (the number 18).

Cryptography uses mathematical problems that have yet to be solved. Need to convert a message (file, image, text, audio) into a number, encrypt it with maths, then send and decrypt.

Ideal function properties to mathematically encrypt your message:

- One way - easy encryption, impossible decryption
- There are no better way of solving it than brute force (e.g. what is a prime number when you cube it & the last two numbers are 27)

Perfect forward security: even if everything has been logged and attackers have worked out a key, attackers can't decrypt the older messages. See Diffie-Helmen's use of session-keys (above) for an example.

RSA - **R**ivest **S**hamir **A**dleman

https://www.openlearning.com/unswcourses/courses/sec-21t1/activities/learnhowrsaworks/?cl=1

1. pick two primes: by tradition we call them **p** and **q**
    - Modulus, **N** needs to be bigger than the number you want to encode
    - we keep them SECRET!
2. calculate the modulus: **N** = p * q
3. calculate a number **m** (mathematicians call it the "Euler Totient")
    - **m** = (p-1) * (q-1)
4. pick one public key - **e**
    - Must be co-prime with **m**
    - List out factors of **m** (what numbers divide **m** evenly?)
    - List out factors of **e** you chose
        i. Make sure the only common factor is 1
5. compute the matching private (decryption) key - **d**
    - **d** is the inverse of **e** mod **m**.
    - i.e. throw into wolfram "inverse **e** modulus **m**"
        i. replacing e and m with your numbers

Using RSA

To encrypt longer messages, either get a longer N or break message up and encrypt each part individually

Encryption

- Convert data into numbers **x** (they'll tell you how)
- encrypted message = **x ^ e** modulus **N**

Decryption

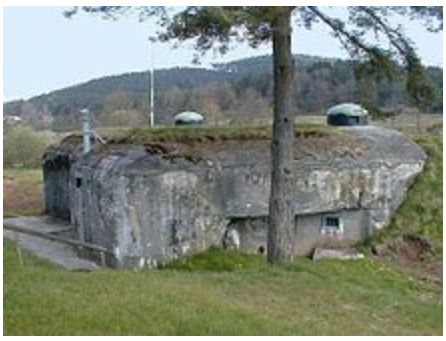
- Given encrypted text **y**
- decrypted message = **y ^ d** modulus **N**

Week 4 Core - Tuesday 9th March 2021

A question: Suppose working very busy. Suddenly get an email from hacker.
Wat do?

Day 0: Just discovered you have been breached, a bunch of files are gone and have been encrypted

The Maginot Line [wikipedia]



"The Great Wall of France", worked as well for the French as it did for the Chinese. Essentially a very very expensive wall defending France from Germany.

Took only 5 days to bypass the Maginot line through the Ardennes forest by violating the neutrality of Belgium, Luxembourg and the Netherlands

Moral of the story: France expected Germany to **attack at their strongest point**, instead Germany went through Belgium and the **weakest point** of the wall.

"Only as strong as your weakest link"

M&M

Build a hard shell outside, assuming no one gets inside (the soft part).
Is the wall really safe?
Wall or "Wall" example: Firewall, Iron Dome, Trump's wall, Great Wall China, Berlin Wall, Jail's walls, Border Security, Hadrian's Wall, *Attack on Titan (great example!)*…

Moral of the story: People want to solve simple problems by building a wall, they don't want to analyse the problem (let's fix the easy problem and build a wall)

Alternatives to walls:
Castle/fortress:

- Problem is a single point of failure, if they get inside the wall, you're doomed and its a massacre like Caesar massacred the Gauls in Alesia.
- Attackers send fake message to the boss saying like "All good here"
- **Second moral of the story:** Walls give a false sense of security and are useless if the **weakest point is the people**
  -
- Castles have multiple layers of walls -- **Defense in depth** (setting up multiple barriers so if one falls it won't be game over). Still better than a single wall (though more expensive).
-
- In early castles, attackers attacked the corners of the rectangular building, as they are easier to shatter. [The Roman architect-engineer Vitruvius (of Vitruvian Man fame) suggested building round walls/towers in his seminal 10 volume work on architecture: *De Architectura*. However this knowledge was lost in the Dark Ages and later independently rediscovered.]

Insiders: Trust based on Profession

Top: Firefighters, Nurse, Doctor, Teacher, Cop*...
Bottom: Politician, Banker, Lawyer…

**Cops are ranked differently in different countries, in Australia they are ranked top*

We trust them based on whether they work in your interest or their interest - **Selfishness**

Insiders (traitors) in wall are actually the most dangerous and <u>most rarely planned for in security</u>
e.g. Firefighter sets a fire, Nurse kills a baby, Cop participates in crime, cuckoos…

Who watches the watcher?
Google has insider teams which specifically deal with insider threats.

Horror Movie: Monsters are insiders, you lock yourself.

Motives for insiders: Greed, Get blackmailed/manipulated, Hate the company/personal reasons ,

How to deal with insiders:
(Promptly)

- Detect the insiders
- Expel the insiders
- Least-access privilege
- Encourage people to report traitors
- Separation of powers

Whistleblowers: a person who informs on a person or organization regarded as engaging in an unlawful or immoral activity.
Motives for Whistleblowers: Moral obligation to undermine the company

Advice for you if you want to whistleblow in the future:

- Read the whistleblower's handbook
- Regular sequence of events once you whistleblow
- **How organisations deal with whistleblowers:**
    - Discredit the whistleblower
        - Attacks anything about the whistleblower that might be slightly imperfect

        - Viciously harass / punish the whistleblower
            - Why? Deterrence + retribution (vengeance)

Secret Agent:

- The watchers (inner wall of the sanctum) that keep the country safe
    - May sometimes not be subject to the law of the land as ends justify the means

What happens when Secret Agent goes bad?:

- Loss of infrastructure in an entire country as the agent would have access to very sensitive knowledge as a result of the trust put on them
- Cambridge 5
    - Agents that had studied at cambridge university. Ideological reasons
    - Some were influenced by money

Weekly Human Weakness - Corruption

Conflicts of interest: forces acting are not completely balanced

Self Interest vs Interest of Duty

Conflict of interest itself is fine, it's how you deal with your conflict of interest that can cause problems.

Example : if the police resource allocation happens due to someone knowing someone in the police that is considered corruption if it was unnecessary (good people can become insiders)

Everyone has duty to ourselves, duty to the profession

Example of duty to profession: politicians making decisions that are right, even though not in their interest, at the expense of losing the next elections

In NSW, ICAC is Independent commission against corruption

Red Tape: there's something stopping me from achieving a goal and I want to get rid of it

_____

Ended the lecture with book show & tell, with the following volumes (this image has been posted on the homepage too, but was scrolled into the void in no time)


UNSW
Ext 56000
BEN MACINTYRE
A SPY AMONG FRIENDS
KIM PHILBY AND THE GREAT BETRAYAL
CROWN
DINA LANDSMAN
RYLE KRIDO
HISTORY LESSONS
BEN MACINTYRE
THE SPY AND THE TRAITOR
G. K. CHESTERTON • Father Brown Stories
WORDSWORTH EDITIONS
PHILIP ROTH THE PLOT AGAINST AMERICA
VINTAGE
THE EMPEROR OF ALL MALADIES
SIDDHARTHA MUKHERJEE
THE CODE-BREAKERS
DAVID KAHN
SCRIBNER

Week 5 Engineering - Monday 15th March 2021

Scenario: DEEZNUTS

"Imagine it's the 1700's and the telephone has been invented..."

3 audience participants, two as the telegram points and 1 guy going between them

Person (1)----------person (2)

This is a telegraph line for a bank

- Person (1) wants to transfer money to Person (2), and tells the messenger to ask "is there a person with the name and number in your system?"
- Person (2) receives the message, and asks for a confirmation from Person (1)
- Person (1) receives confirmation from Person (2), transfers the money and sends a confirmation of transfer to Person (2)for n

But wait! Richard says there's a problem:

Listed Problems (by audience + R.B)

- Middle-man attack (intercepts message in between the line, and pretends to be the messenger confirming transfer, middle-man gets away with money with the real details he scraped from the line)
- A man-in-the-middle attack, is an attack that places the attacker in the middle of two hosts that think they're communicating directly with each other. A common man-in-the-middle attack is a session hijacking or cookie hijacking.
- Another way a man-in-the-middle attack can be established is a rogue access point attack.
- A final man-in-the-middle method will cover is called an evil twin. It's similar to the rogue AP example but has a small but important difference. The premise of an evil twin attack is for you to connect to a network that is identical to yours.
-

- Solution? E N C R Y P T I O N . We want know that the message has integrity (i.e. the message hasn't been tampered with plus is coming from the right person)

This is an example of the Two Generals problem, where you can never really confirm that the message from someone has come from someone, given that there is a third party involved. This has been proven mathematically, but we have lots of ways to functionally get end-to-end security from Person (1) to Person (2).

TL;DR: Exercise to show that we need integrity (the privacy one)

Hashing: More than just delicious fried potato

Hashing is getting data of any source and summarising it as a short fixed length string. We can use it like a fingerprint to identify data. Hash functions can take a file and turn it into a hash (e.g. Checksum). This can be used for a variety of things, most commonly used for encrypting plain-text sensitive information, so in the event of a database leak, the data is useless for the attacker (theoretically)

Q: But like, why use hashes? Why not just compare the original files?

A: Hashes are faster to send and compare, as opposed to the original file.

Good General hash function properties:

- Unique objects may be hashed to the same value (collision). Where objects have different values in attributes that we want to distinguish between, they should end up with different hashes.
- We want even the tiniest change to create a vastly different hash, so we can use the hash function result to detect if something has changed the file or not. It can be

a good way of checking the integrity of files cost effectively, given that you have a good hash function
- Bcrypt is the current industry standard for hashing.
- Can be stored regularly without worry of compromising the original file
- You want the things you are trying to distinguish to have different hash values, so collisions only occur between things you don't care about

Good Cryptographic hash function properties:
- 1-way (cannot recover original file given the hash value) AKA **pre-image resistance**
    - No faster way to find file that hashes to a certain value than to try all possible files
    - Works as information is lost through hashing producing a "summary" which can build to multiple things if the method is reversed
    - End result is 'pseudo-random' i.e. end result has no relation to the data it represents, but can be validated when checked that this is the correct input to produce the hash
- Collision resistance:
    - It is computationally infeasible to find two inputs which have the same output.
- Second-preimage resistance:
    - Given one object, cannot find a second object which gives the same hash value as the first object
- Avalanche property:
    - A change in 1 bit of the input should cause EACH hash bits to flip with 50% probability

Examples of Hash Usage:
- Shazam compares a hash of the currently sampled audio across hashes or other such fingerprints in their database.
- Digital Certificates for websites (to confirm a website is legit and not a spoofed site)
- Storing hashed passwords: even if attackers have the hashed passwords, they need to enter the unhashed password in order to produce the hash to log in
    - **CRC (not cryptographic)**
    - **MD5 (cracked in the 90's)**
        - 128bit hash
        - MD2 made in 1989

    - MD4 made in 1990, weaks found in 1991, broken in 1995
    - MD5 made in 1991, weaknesses found in 1993, broken in 2005
    - MD6 made in 2008 (256bit, applied to be SHA-3 and failed lmao)
- **SHA (NSA made)**
    - SHA-0 made in 1993 (not secure, broken literally right after release RIP)
    - SHA-1 made in 1993 (160 bits, not secure, broken in 2005 ish)
    - SHA-2 made in 2001 (224-512 bits with 64-80 rounds of hashing, sus)
    - SHA-3 made in 2015 (**NIST approved!** 224-512 bits)
        - Different structure to predecessors
    - SHA-kira made in 1977 (Produces absolute bangers, very secure)

“Hash is swiss army knife” - Richard Buckland

Hash-->hash-->hash-->hash-->hash-->hash-->hash-->hash to make it more secure, but it costs a lot of computer energy to hash data so many times, esp for large sets of data

Difference between hash and cryptographic hash is that a cryptographic hash should be:
- Preimage resistant (given output, can’t work out input)
- collision resistant (given the hash function and you cant find two messages that give the same hash)
- second pre-image ‘collision’ resistant (given the hash function and one input, you can’t find another input that produces the same hashed output)

Richard stole microsoft t-shirts and laughed at them

Hashing: Socially Distanced Hashes for Integrity

EXERCISE TIME!

Let's imagine everyone is summarised by their birthday by a hashing function:

> bday(person);

The output of this function is between 1-365. How long will it take to get a collision with this hashing function?

The results:
16 aug
2nd feb
21 may
19 sept
10 july
20 nov
6 mar
29 oct
1 jan
30 aug
9 jan
1 sept
28 jun
7 feb
9 aug
4 mar
15 may
27 nov
27 dec
15 jan
13 aug
3 aug
27 jan
20 aug
24 may

23 aug
27 sept
12 aug
30 feb
21 aug
1 jul
13 feb
4 jan
26 apr
4 feb
1st jan -- HUZZAH!

Took 33 entries to get a collision ish

Birthday attack:

- Group of 23 people -> approx. 50% chance of having a birthday collision
  - https://en.wikipedia.org/wiki/Birthday_problem
  - Method to find:
    - Find the 50% probability that a birthday collision is not found
      - 1 * 364/365 * 363/365 * … * 343/365 = approx. 0.492703
    - Therefore, chance to find 50.7% probability of collision requires (365-343) + 1 people => 23 people
- Average runtime is SQRT(N), where N is the number of possible hashes
  - Thus, 128 bit hash will take 64 bits of work to crack using birthday attack.
  - Therefore, 256 bit hash ensures even birthday attack requires 128 bits of work
  - E.g. sqrt(365) = 20
- https://www.youtube.com/watch?v=ofTb57aZHZs , <- recommended watch to understand more about the birthday attack

Now applying this to hashes, why do we care about collisions?

Imagine the will of R.B.

R.B. hashes a document of his will bequeathing his money to MSF. R.B. sadly used 'joke-hash', a very insecure hashing function to act as a proof of validity of the document.

Eddy (our l337 haxx0r), wants to take R.B. 's money, and makes a fake will identical to the original document.

Eddy then proceeds to hash them both together until he finds two with identical hash results. Huzzah! Eddy makes away with R.B. 's money and runs off into the sunset.

In this example, the two documents eventually collide in the same hash value due to the bad hash function, rendering hashing useless for validation purposes.

Ideal Hash Function

- Given a hash(x), it is hard to figure out x from the hash output
- Tiny changes = vastly different hash, to detect if file has changed
- Collision resistant
    - Message authentication code
    - Protects the integrity of message
    - Example: Send A $1 million, banana
    - Attackers can intercept hashed messages
    - Attackers will not know the word "banana", cannot fake pre image messages without the code
    - Can also use a counter (number series so it is unique), stops replay attacks

Man-in-the-middle attacks

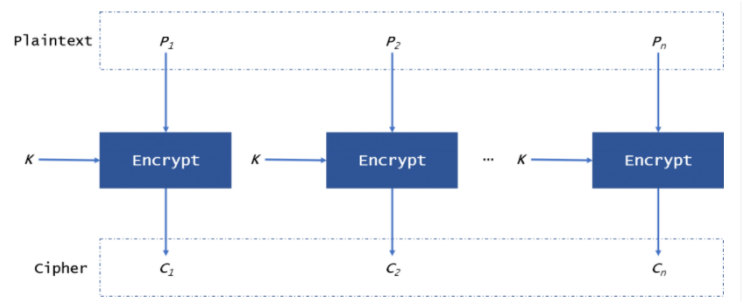
Example of the Two Generals problem, hashes are the best thing we have so far to solve this problem even though we know the Two Generals problem technically has no solution

Example of bad hash IRL:

"Here's a video of me hashing a hard drive, and the hash is a perfect record showing that I haven't tampered with the drive at all through the analysis, and that I haven't added or removed files on this drive ;^). I did it with the power of MD2!"

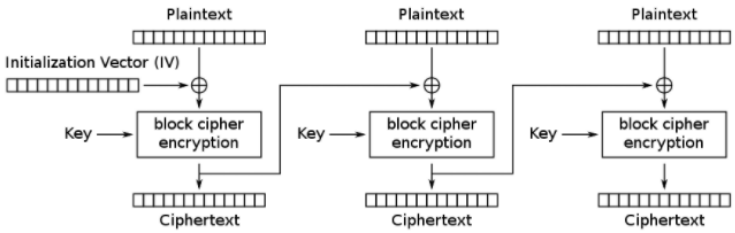
Block Modes
[https://www.highgo.ca/2019/08/09/the-difference-in-five-modes-in-the-aes-encryption-algorithm/]

1. ECB - Electronic Code Book
    a. Vulnerable because blocks can be repeated and it's relatively easy to tell the edges of the blocks
    b. Partition and encrypt each partition, then join partitions together again

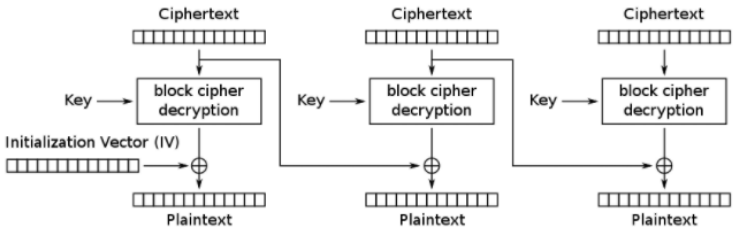


    c.

2. CBC - Cipher Block Chaining
    a. Partition plaintext and XOR it with the previous partition's encryption
    b. First partition is XOR'd with some key before being encrypted



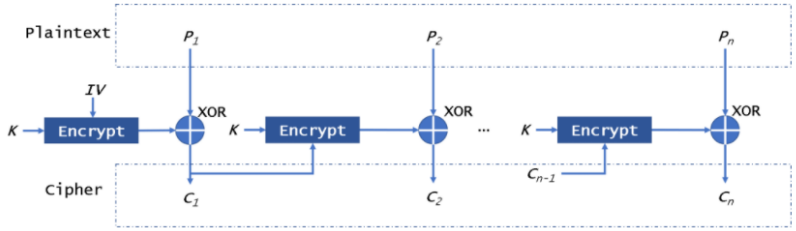
Cipher Block Chaining (CBC) mode encryption



    c. BC_encryption.svg Cipher Block Chaining (CBC) mode decryption

3. CTR - Counter
    i. Encrypt the initialisation vector (IV)
    ii. Then XOR with plaintext, which makes ciphertext
    iii. Repeat

iv.


Plaintext
P_1
P_2
P_n
IV
K
Encrypt
XOR
K
Encrypt
XOR
...
K
Encrypt
XOR
C_{n-1}
Cipher
C_1
C_2
C_n

Week 5 Foundations - Tuesday 16th March 2021

Privacy: Sssssshhh, it's a ***secret...***

The Time Richard Paid for Parking:

Richard drove today! Unfortunately he still has to pay for parking through CellOPark.

He enters in data:

> uni (optional)

> staff ID (optional)

He tries to go without it, but the app forces you to write something.

He proceeds to write gibberish, because they already have his credit card details. ***Privacy!***

R.B.

TL;DR: R.B. warns us about the danger of our private data and what it can be used for, and how we just give it out everywhere.

Security Everywhere, What Could Go Wrong?

On a cold day in the UK, a child was abducted at a store in London. Two teen girls were attempting to kidnap a young child, and had planned it out well.

Police officers used the various CCTV activities to figure out where the girls were travelling within half an hour of the missing report, and the girl was recovered.

The two teenage girls were arrested, and were suspected of wanting to kidnap the child for nefarious purposes.

Many other cases have been solved through the use of security cameras being able to piece together that information and to catch the people who have hurt others. Obviously it's great to collect all of this data, as long as it stays in the right hands, right? Well...

What Goes Wrong:
All the data that we collect is a huge target, it can be the target for attacks, and if leaked could be used by anyone for whatever purpose. Remember, data doesn't have an agenda, but people do!

Problems with mass data collection

- The purpose the data is collected may not be good (data could be manipulated)
- The purpose the data is collected may not be the only purpose they claim it's used for (possible violation of given consent)
- Data may be misused / leaked by accident / stolen (*"Who will watch the watcher?"*)
- Type 1 Type 2 Errors

ELI5: Data Collection

R.B. tries to be honest to his kids. He tries to exemplify this to his kids. But. There's one scenario where R.B. is happy to lie; divulging personal data to anyone. R.B.'s kids are confused. Why be honest anywhere *but* the internet and on data forms?

R.B. says: *"Sweetie you must never ever ever **ever** tell a lie. **Except** if you're typing data on a form online, then lie lie lie! Lie to the government lie to the app developers, lie to the nice lady on the street wanting donations, lie lie lie and never tell the truth about yourself online >:("*

R.B. then proceeds to explain to his kids that you can't always trust people with your data, and that one incompetent governmental person can result in all your personal information being available to the public.

TL;DR: trust nobody ~~red~~ data collection sus

Identity Theft: We are <u>ALL</u> Richard Buckland on this Blessed Day....

One day R.B. loses his birth certificate. So, he goes to the Registry of Births and Deaths to get a new one.

He had nothing but secrets about his parents and a drivers licence and he suddenly gets to have a new birth certificate with a new pretty photo of him!

He was then able to use this new birth certificate to get a completely new passport.

But what if it wasn't Richard Buckland? What if it was a man trying to become Richard Buckland legally to take his legal assets? You'd be surprised on how much data you leave on the internet!

<u>R.B. Tips!</u>

How to detect ID theft:

- Set up notifications for bank activity etc (send alert whenever privileged actions are carried out in your name)
- Request credit history report to detect anomalies
- Request all the information a company has on you to see if someone else is using your identity

"But Richard, I have nothing to hide!"

What if someone doesn't like you? They suddenly have access to all this data to make your life really hard. In the example above, all the data you don't feel the need to hide could be used to completely steal your identity. Extending this concept, **<u>surveillance</u>** changes the way that people act. Why?

Privacy-induced behavior change depends on how high you are on the primate tree. Higher primates behave more differently depending on whether they are observed. Be monke. Stonke. doge 💀💎🙌. Humans are especially prone to this behaviour, even if there isn't anything particularly sensitive on what you're doing.

A great example: The Panopticon. Panopticons are donut shaped prison design where you're always being watched. Even if you're not being watched, your behaviour will change if you think you *could* be being watched

Surveillance relies on a lack of faith of people to justify data collection. After all, if there is no surveillance someone could pretend to be R.B. and take his identity (and will with a sizeable inheritence) But can we trust people?

Would I Lie To You? (If they're R.B. kids, then yeah probably lol)

Privacy is a telegraph pole, with three parties balancing it up. Each has reasons for either wanting data or giving data or lack thereof:

Governments/Police:

- Stop the spread of covid
- Anti-Terroism
- Taxes
- Census
- Catch criminals and stuff

Companies

- Selling your data
- Outmaneuver their competitors

Our individual motivations

- We want
    - Personalised healthcare
    - Crimes to be caught
    - Democracy
- We don't want
    - Identity fraud
    - Leaked data (from incompetent dataholders)
    - New Zealand secret service /__(👁_👁)__/\
    - Robo debt
        - Literally hounded by the government for a bad system

It's more important than ever that organisations and data holders need to be honest with what they do with your data, and if it ever gets lost/stolen.

At the time of collection of any data, it might have been in safe hands. But people change. Organizations change. They could collect it for one purpose, but then organizational structures could change, and the data used for something else. Data never gets deleted (mostly).

Erol's example: uploaded photos in 2009, but now those photos are being used to sell facial recognition tech to law enforcement.

Steps to being dumb with data:

1) There's something to measure, might as well measure it :shrug:
2) We have all this data, let's use it and analyse it!
3) We've analysed all this data, let's use it!

Forward secrecy/security

- Just because you have nothing to hide *now* doesn't mean you won't have anything to hide in the future. For example: Rwandan genocide, people disclosed their ethinicity in their ID cards; Journalists who reported on the Afghan files may regret providing the government with a long fingerprint on their ordinary behavior, their social connections, etc.

// DO **NOT** DELETE THIS LINE I'm going to hash this line to maintain its integrity

    -

Week 7 Engineering - Monday 29th March 2021

How do you know what is rEaL

- Computers just process/send/manipulate data. How do you know you're talking to the right person? — the problem of authentication.
- Very hard for computers to "know" about the outside world, then.
- **Stories**
    - Alice in Wonderland "don't wake the king - he's dreaming us" -> how can inside the dream interact with outside the dream?
    - Transistors - CMOS cookbook - interaction between the information world and the physical world
- We generally get authentication right in the real world, but it is hard to **know** these things.
- Generally use "factors" to authenticate people —

Authentication

- **Authentication Factors:**
    - What you are = a fingerprint
    - What you know = secret
    - What you have = to verify a physical attribute like a fingerprint, you may require a transducer
    - What you can do = e.g. a skill

- Once they find the secret they could crack everything, only needed one factor to crack all factors, knew every secure factor making it no longer secure
    - All authentication is really just one factor
- Two factor authentication on phone: what you know and what you have (your phone), send a SMS to your phone to authenticate
    - Someone could forge a sim card - that's just a secret
- Authentication: daily, visible action completed all the time, logging into computer, phones, checking into work etc .

    - Visible, measurable, tangible action

AUTHENTICATION PROTOCOLS - Fall Back on when confused

- Authentication products - lots of jargon, "goobledygook"
    - vendors "sending dreams", probably only caters to one error
    - Offering a miracle for one error but a nightmare for the other
- There'll always be **type 1 / type 2 errors** in authentication
    - E.g. getting money out of bank account, two errors: bad guy getting money out of my account OR I can't get money out of my account
    - When t1 errors go up, t2 errors will go down and vice versa. Finding balance is difficult
- Authentication ties a request to an **identity (and this is a DIGITAL identity)**
- **IDENTITY** never going to be an actual real world identity - a piece of text (phone number, email address, TFN)
    - Can only tie a request to a piece of data, can never truly authenticate that it's your mum talking to you
    - "They have access to the phone number"
    - Phone/email not a terrible proxy for identity - you have usually have one phone, check your email regularly and keep it close
    - BUT email address can be owned by multiple people, read by vendor/supplier, company email, shared password etc
- Visibility of authentication has 3 properties
    - Annoying - pressure to make authentication less annoying
    - Attract funding - "space age, scifi" "more advanced and agile than others - fingerprint scans, facial scans, biometrics" justify big spends. People like spending on visible things than intangible things
    - When things are visible, and fail in the visible way - pretty good for security BUT bad guy always try to do authentication silently
- Authentication vs authorisation
    - Authentication = linking identity in a certain way / Make sure you are what you are
    - Authorization = giving permission to undertake an action
    - Usually authenticate yourself by logging in (get a **token** to do things as whoever you've logged in as e.g. GitHub personal access tokens), token authorises you to keep having access to a resource
- Authentication - timing ,duration, frequency

- Authenticate constantly (eg. each keystroke) -very annoying
- Usually authenticate less frequently and trusted between times
- Authenticate at the beginning and think it's ok then on
    - Have authentication tokens at critical times but most of the time just have authorisation to allow you to do anything for a while
        - Privileged activities might need more authentication but most activities just rely on the initial authentication of logging in
- M&M property - protect the border - assume everything bad is on the outside
    - Temporal M&M, outside thick side is authentication then on inside it's just authentication yum chocolate soft

- **Authentication depends on Parties involved**:
    - Look at all these different authentications that have their own challenges and protocols:
        - client authenticating server (e.g. you logging into gmail)
        - server authenticating a client (site telling browser that it really is amazon.com)
        - peer authenticating other peer
    - A danger of authentication: sometimes it's easy to forget that authentication is like a web spun between all different parties involved. Don't focus on just one connection between 2 members and think that's all the authentication that is needed
    - Most of the time, there are multiple parties involved in a system and they need authentication to one another, however we have the tendency to only focus on authenticating 1 or 2 parties.
- **CHECKLIST ^**
- **Penn and Teller episode**

Authentication Protocols

- Challenge that you will reveal information by authentication but want to prevent replay attacks from occurring
- Only way to authenticate in most cases is to convince we know a secret
- Challenge response: mum what did i eat for dinner last night, answer changes all the time, preventing replay attacks
- Proof of liveness: check that the person is actually there

E.g. banks need to use a card to log in, get into building, go pee → need to take it everywhere, proof of liveness as if the card is there you probs are too

- **S/Key**
    - Use the rightmost key first (server needs to remember this key), then the key just before it (server can verify hash of key is the last key), and so on, each time server just needs to remember the last key and check that the hash of the each key offered matches the most recently used key.

      Uses hashes
      Had to type password in the clear if sniffing
      Password that can only be used once (one time password)
      Start off with an initial secret (could be your own password to log in) → hash it with a good hashing algorithm e.g. SHA3 → take hash and hash it again with SHA3 write it down → hash again → hash again
      hash(hash(Hash(Hash(Hash…..))))Keep hashing the hashes
        - Have a list of all hashes down but the computer system only lets you to use them all one time
        - Computer can check that hash is correct instantly, but hacker cannot go backwards with a hash
        - Cons is: Can be used a finite number of times, finite, but clever
- TOTP: temporal one time password

  Time orientated, doesn't use counter for no. of times used but a counter that updates every 30sC
  How can time-based authentication be exploited?
    - Attackers can brute force to race you to gain access if they see the first few characters of your verification code
    - If the website accepts multiple authentications then they can get the same token as you if they take your code and verify within the same time-frame
- HOTP: hashed one time password

  Have a counter → every time you use it counter increases by 1 → get counter and hashes using HMAC (keyed MAC, you and server knows the shared secret)
  Allows for infinite number of times to use password unlike S key
  Google authenticator uses this

- Can use RSA backwards, encrypt a message using private key and then send it to someone who can decrypt it with my public key → good way to authenticate i.e. see if you know your private key

Signatures

- Both physical and digital signature should ensure authentication and integrity
- Signature should authenticate you as you and ensure the integrity of the document (that it's been read and agreed to and it won't be changed after you sign it)
- You can use RSA to do ^:
    - Using RSA: Can encrypt the entire document using private key (creating a ciphertext) and then attach it to the document, and that would be a signature!
    - To verify: decrypted the ciphertext using public key and it should return the same document the ciphertext is attached to. This works because only you know your private key. Provides authentication and integrity of document.
    - Non-repudiation: You can back out of any contract by saying "oh no i lost my private key! Any hacker could have used my private key!" and now all the documents containing your signature is rubbish :)
    - Problem 1: The person writing the contract has control over the wording. You don't want to use your private key to encrypt something other people have control over like that because they could be planning an attack (Someone could reverse and find your private key). General rule: don't sign document other people have produced.
        - Prevention: we usually hash the document before the encryption (signing).
    - Problem 2: What if the document is bigger than one block in the RSA (RSA only does a block of a certain size). Now you have problem with block modes
        - Prevention: we usually hash the document because it makes it a fixed size and then encrypt the hash using the private key.

- OAUTH

  Single sign on which spans across different organisations i.e. would you like to sign in using google account?
  First authenticates by logging into google then it sends authorisation tokens to other websites to allow you to do things using the google account

Google, facebook, amazon, microsoft, twitter and, and, and, and
Convenient, annoyance massively reduced but what are the trade offs?

Authentication Attacks

- Fallback attack:
    - If you fail to authenticate too many times, server will degrade to lower authentication standard (e.g. WIFI uses WPA if you do not have WPA2). Bad guy can force the fallback to be used.
- Password recovery:
    - Attack the password recovery workflow (social engineering, weak fallback questions whose answers can be recovered via social media)
    - Set up a email made ***only*** for password recovery, make it really strong key, don't use for anything else, if someone compromises this then attacker has everything
- SIM cards:
    - Reddit attack in 2018: 2FA over SMS, bad guy bribed AT&T employee for $75 to gain access to reddit employees' SMS deets
- Interfaces and transducers:
    - Hardware written by third party, trust foundation is wider. Insiders and backdoor. Risk minefield, higher chance of failure.
- Session Hijacking:
    - Steal your authorisation token
    - Shown in the youtube video in lecture, (https://www.youtube.com/watch?v=xaOX8DS-Cto)
- No Lock-Out
    - No limit on how many times you can enter an authorisation token so you can just brute-force until success

Week 7 Foundations - Tuesday 30th March 2021

Ransomware

- Ransomware holds data in exchange for money
- Would stop if EVERYONE does not pay
- They could also release your data

Identity Theft (is not a joke jim)

- Impersonation of your data
- Take credit cards, loan, etc.

Cluttered Data

- Do not hoard data you do not need, it is an opportunity to be stolen and misused

How to delete data in secure way

- How to get rid of PC (ideas only)
    - Take all hard drives out
    - Drill holes
    - Soak it
    - Pass magnet over it
    - Cut in half, bin in different locations
- Dangers
    - Overwrite is not perfect, it leaves a thin outline of previous data behind, this is how recuva works

Data Wants to be Free

- They are never talking about their own data, they are talking about other people's data
- Mark Zuckerberg does not want his privacy corrupted, but he is okay with other people's data exposed

Deadman Switch

- Must be holding onto the switch for it to be disarmed
- This is implemented in trains in case something happens to the conductor
- Bypass: They placed a brick on the pedal

Bombs

- The US and Russians were concerned about launching on each other
- They kept thinking that the other was launching on each other
- The implementers were worried about not being able to launch the bomb when they need
- Others were worried about launching the bomb when they do not need

Pyramid hierarchy

- Gives one human all the power, strength of 1000
- The largest disadvantage is that this one person can become corrupted

Spreading out power

- Give people with power a weakness
    - Head of department can only be head for a limited time
    - Police has power over citizens, but citizens can vote officers out, etc.
- Centralised vs decentralised
- Use checks and balances

Red Tape

- Red tape is an idiom referring to regulations or conformity to formal rules or standards which are claimed to be excessive, rigid or redundant, or to bureaucracy claimed to hinder or prevent action or decision-making.

Dual Control

- Two people must agree for it to be launched
- Reduced threat of insiders and corruption
- Example is two keys in missile launches

Book Keeping

- Every transaction is entered twice in two different books by two different people
- To fiddle with the system, two books must be corrupted
- Can cross check to detect tampering

Development Methodologies

- Agile vs Waterfall
- Waterfall (traditional): spec out everything first, implement all at once, evaluate at the end
- Agile: iterative process of development involving sprints with shorter term plans and dev cycles

R.bucky is a hoarder

R.bucky used to be really into bush restoration?

Data Breaches: All Your Base Belong To Us

Data, much like gold, is very valuable.

And much like gold, people love doing heists to steal data which can be sold or used for much more than exchanging for goods and services! The ownership and exchange of data has become a very profitable business for cyber-criminals, with examples of this being:

Ransomware - With helpdesks, 24/7 customer service, and responsive agents, the people running the malware operations like WannaCry and co. offer a business-like experience with your data, forcing you to pay money for the recovery of your data.

Blackmail - Whether you're a cheating partner in the Ashley Madison database, or a multinational company with a secret business plan, cyber-criminals can blackmail people at a whole new level with how many valuable secrets are stored online in databases.

One especially egregious use of data gained from data breaches is identity theft.

Connect 4: Value-adding the pieces of your online-presence
Despite how careful you may be (read: incognito mode), you leave a profile of behaviour on whatever system you have entered your data into, which can be used to track you and everything that you do.

Despite many claims of companies and governments anonymising data, it's surprisingly easy to de-anonymise data if you try hard enough (after all, the data hasn't been generated in a truly random manner, so given simple parameters like birthdays, important dates, and other identifying tidbits, you can analyse the data that fits the pattern you're looking for).

Case Study: Vanessa Teague and the Medicare/PBS dataset
Anonymous medicare data wasn't and vanessa proved it ayyylmao
https://www.theguardian.com/australia-news/2020/mar/08/melbourne-professor-quits-after-health-department-pressures-her-over-data-breach

DaTa LaKeS:
A data lake is a collection of data stored in it's raw format. As humans, we love to hoard items, and collections of data are no exception. To maintain the best forward-secrecy, it's best to just delete older data as a habit; the most secure data is data that doesn't exist!

Openness: Data Wants To Be Free!

Or does it? Data wants to be free… unless it's your own data then obviously you'd wanna keep that protected, right? This complicated push-and-pull of data flow is something that many data analysts pore over, and we have many examples of what people have used open data to do:

- China Social Credit system
- FOI requests
- WikiLeaks

Submissions for what
I think it's submissions for objections to privacy law changes lol
Privacy law chan u w u

Proposed laws will be put to the public and you can give your opinion on those laws - the submission. Contributing via submissions allows for the creation of better laws (laws which reflect the technical reality as opposed to just what lawyers see).

Nuclear Bombs: The Weight of Decision

Book Rec: Command and Control by Eric Schlosser

Command and control:
Pyramid control:

- Pro:
    - One person has the strength of a million
- Con:
    - Strength of a million person has the brains of one person
    - Communication overhead of moving commands from the top to the bottom

According to Richard Buckland, its good to have centralized systems of government during times of urgency where important decisions need to be made quickly like wartime. Otherwise decentralization is better.

Red-tape: Something that is stopping someone in power from doing what they want to do

Week 8 Engineering – Monday 5th April 2021

Podcast Episode 8.1: Encryption on the web - TLS, SSL, HTTPS

Old days on the web

- Very limited encryption
- Everything sent in plain text
- Had to assume no sniffing
- Local area networks setup so anyone could see everything
- Use telnet to login remotely (all passwords sent in the clear)
- Everything was in the clear
- Mosaic browser was graphical
- People could see what you're looking at
- Internet was a delivery service for information
- Interaction with the webpage was also in the clear
- If you wanted encryption, you had to do it yourself
- As serious systems went online - realisation that we need encryption
- Netscape - early browser company
    - Had a famous cryptographic hero
    - Secured info
    - SSL would replace Berkley socket
    - Abstraction
    - Secure socket layer (SSL)
    - HTTP over SSL = HTTPS
        - Independent of user
- First version of SSL in early 90's
- SSL 2.0 1995
- SSL 3.0 1996 (Depreciated in 2015)
    - Had severe problems known in a year or two
- Can trick server to use weaker security
    - Mr nah from picking Netflix movies
    - Fallback attacks deez nuts
- Wasn't a priority to upgrade to better SSL
- NHS used to use Windows XP for way too long
- Microsoft didn't see internet would be a big thing

    - Introduced internet explorer later on
    - Broke the standard because other browsers didn't support certain things
        - Deliberate (as a strategy)
        - Introduced special HTML tags only compatible with internet explorer.
        - You had a different experience to everyone else if you viewed from microsoft browser
    - Microsoft just wanted money
- Changed SSL to TLS (Transport Layer Security)
    - Just so it didn't look like a netscape thing
- TLS 1.0 1999, TLS 1.1 2005ish & stayed around until recently (2020ish)
    - Turned out to be vulnerable as well
    - Accepted use of protocols that were broken (RC4, MD5 etc.)
    - Stupid cheap browser
    - Fallback attack
- TLS 1.2 2008
- TLS 1.3 2018
    - Everyone should be using this now
    - Became very tough
    - Throws out lots of algorithms that are weak and compromised
- Used the intel holographic sticker to advertise on computers

How does HTTPS negotiate and set up connection

Properties

1. TLS ClientHello contains Server Name Indication (SNI) / handshake phase
    - Required to serve multiple HTTPS sites from the same IP
    - Things u want / properties
        - C - Confidential (actual data should be invisible)
            - Ciphers
            - Encrypted
                - The actual data is encrypted (the rest of the URL)
            - Visible / not encrypted
                - IP address
                - Port number
                - Volume and timing

            - Who you're talking to - e.g. google.com
        - Symmetric cipher (Stream or block cipher)
            - Most usually a block cipher- e.g. AES
    - I - Integrity (don't want people messing with data)
        - Series of blocks sent in this connection over these sockets
        - End of block - message authentication code (MAC) attached
        - All block ciphers in TLS 1.3 - ciphering mode does integrity checking
        - Don't want people to fiddle with what we're saying
    - A - Authentication
        - Forward secrecy
        - Browser authenticate that its talking to the right web server
        - Https, tls doesn't normally authenticate the other way, the web server doesn't know your browser
2. Communicating
    - Use cryptography

Hand shake (5 Steps)

1. Client connects to server. (client doesn't know what protocols server can and can't support)
2. Server responds to say which protocols/cryptographic primitives will be used (normally one encryption and one hash to do authentication and confidentiality)
3. Server then sends a certificate to the client and this certificate authenticates the server through a process called pki
4. Client checks that certificate is valid
5. Client generates a temporary session key that the client and server will use to communicate. There are 2 approaches to generate this
    a. Run a random number generator and encrypt the number with the public key of the server, then send the key to the server. Only the server knows the private key.
    b. Diffie Hellman- next podcast
    - There's a MAC associated with this- can be attached at the end of every block
    - Authentication happens when we have a certificate. Only person named in the certificate can understand

- In security don't trust anyone or anything
- TLS encryption uses a symmetric cipher
    - This is faster, even though setup uses asymmetric cipher
- TLS 1.3 is savage as it doesn't accept anything even potentially insecure, e.g. cipher block chaining as there was an attack discovered 'lucky 13'
    - Only accepts protocols that accept perfect forward secrecy- like variants of Diffie Hellman
    - Only mode uses is a variance of counter mode, GCM, CCM

Understand these:

- Fallback attacks
    - Tricks computers to abandon newer more secure versions of operations to older less secure ones
- Poodle attacks (SSL 3.0)
- Heartbleed
- Don't ever roll your own crypto

Attacks

- **Poodle attack (Padding Oracle On Downgraded Legacy Encryption)**
    - **Overview**
        - Is a vulnerability common to SSL v3.0
        - Is Man in the middle + fall back attack
        - whereby you force a session to fall back to ssl v3.0 and then go to town
    - **How does it fall back?**
        - You just interrupt the client to server handshake enough times that it forces the session to connect via ssl v3.0
    - **Then?**
        - if cipher suits RC4 or block cipher in CBC mode
        - Attacker can retrieve partial bytes of encrypt text, and later get full plain text
    - **References**
        - https://www.alertlogic.com/blog/poodle-the-man-in-the-middle-attack-on-sslv3-d62/
        - https://www.youtube.com/watch?v=XxO7L5gHshY
- **Fallback attack**
    - man in the middle attack

    - interrupts connections and thus pushing systems to use legacy cryptographic protocols
    - which are then susceptible to hacking
    -
- **Heartbleed attacks**

    - **Overview**
        - When a client and a server communicate with one another they connect via SSL
        - Servers tend to host a lot of sockets, so it terminates some of these sockets
        - **To prevent the server from closing their socket**, the client sends a message (saying hey im still here) known as a **heartbeat**
        - This data is not encrypted.
        - **Furthermore, these heartbeats can be of varying sizes less than 64kb**
        - And servers respond in kind with the same size eg.
            - client sends 1, server sends 1
            - client sends 64, server sends 64
        - eg. heartbeat actual packet size = 1kb, but says it is 64kb,
        - server responds with 1kb and 63kb of random memory
        - could hold passwords, usernames anything
        - these can be sent in quick succession and numerous times
        - Getting useful information
        - The computer that received the heartbeat request never checked to make sure the request was actually as long as it claimed to be. So if a request said it was 40 KB long but was actually only 20 KB, the receiving computer would set aside 40 KB of memory buffer, then store the 20 KB it actually received, then send back that 20 KB *plus* whatever happened to be in the next 20 KB of memory. That extra 20 KB of data is information that the attacker has now extracted from the web server.
        - This is the crucial part of the operation. Even when a computer is done with information, it persists in memory buffers until something else comes along to overwrite it. Attackers can take that extra 20kb memory and potentially find crucial info.
        -
    - **References**
        - https://www.youtube.com/watch?v=6Sz5wBBXzpc
        - https://www.youtube.com/watch?v=WgrBrPW_Zn4
        - https://www.csoonline.com/article/3223203/what-is-the-heartbleed-bug-how-does-it-work-and-how-was-it-fixed.html
        -

Episode 8.2: PKI, X.509 and the Ministry of Silly Walks

- Go amazon ip
    - How do we actually know this is amazon?
- Use secrets
- Knowing a secret no one else knows could be enough
- 2 different approaches to do this on internet
- Approach 1: Web of trust
    - Crowdsourced truth
    - Talk to enough people that are trusted that can authenticate
    - Decentralised system
    - Used by famous email encryption (PGP)
        - Everyone exchanges keys when they meet
        - Key signing parties
        - Accumulate a whole lot of signatures
    - Problems
        - 1
        - 2
            - Certificate authorities can sign certificates from other companies and create a link of certificates who trust one another that may be malicious
- Approach 2: Pyramidical Structure
    - PKI - Public Key Infrastructure
        - Solves the key distribution problem- don't need to send keys to everyone
        - Anyone can see your public key
        - Use your private key to decrypt
        - But how do I find your public key?
            - On webpage - which webpage do I trust?
            - Which key belongs to which person?
            - Man in middle attack

            - Can promote false keys and trick people!
        - Central system tells everyone what the public key is
            - Insert false key in central system?
            - You prove who you are to certificate authorities and they sign certificates
            - People's incentive is you pay certificate authorities money to sign
        - When you ship browser - **browser has certificates preloaded of which authorities to trust**
            - Use to be able to just pay to be included as a preloaded certificate
            - How do web browser companies know who to trust?
            - In the old days, you pay to be on the list
            - Governments tell you to put them in
            - Browsers put their own companies in
            - Now this certificate list is stored in the OS as well
            - You can see it on your key chain
        - Are certificate authorities trustworthy?
            - No quality control on certificate authorities- there could be clowns running the certificate authorities
            - Once you have certificates in the store, they're all equal
            - Chain of trust issue - A browser's preloaded trusted certificate authorities could trust another intermediate CA which in turn could trust another CA(could be malicious) and so on until it reaches a trust anchor.
        - **Ties domain name to certificates instead of companies**
- It's hard to revoke certificates!Called X.509 certificates because they used to be part of a wider protocol
    - Protocol was never set up to enable easy revokal
    - Used to use revocation lists but no actual mechanism to create or distribute them (especially now because they can be huge and there would be so many of them) z
    -

Episode 8.3: DH, Forward Secrecy and MITM F

- Diffie Hellman Key Exchange
    - Raise 2 to a power and then mod it
        - 2^6 mod 11 =

- This problem is known as the discrete log problem
- DH relies on the one-way hardness of this problem -> can't get much faster than brute forcing with this problem
- We have Alice and Bob who want to communicate securely but they never got together to establish a key

1. On a **public channel** they agree on what number they are modding by a prime (n or p) and a number that they are raising to a power (base or g)
    a. g raise it to some power then mod by n
    b. This is agreed on open channel so eavesdropper knows g and n like they do
2. They each think of a secret number a and b (must be between 0 and n-1)
    - Alice thinks of a (3?)
    - Bob thinks of b (7?)
3. Alice does a secret computation and computes C = g^a modn
    - G = 2
    - N = 11
    - Each going to calculate 2 ^ secret number mod 11
        - Alice: 2^3 mod 11 = 8 mod 11 = 8
        - Bob: 2^7 mod 11 = 7
4. Bob does the same computation for b -> D = g^b mod n
5. Now they exchange their answers -> Eve (eavesdropper) would have to resort to brute force to calculate a and b
    - Eve thinks: 2 ^ something mod 11 = 8?
    - She looks at bobs one 2 power of something mod 11 = 7
    - If n was big enough - the amount of eve would have to do is enormous
6. Alice takes the number that Bob sent to her and raises it to her secret power and mods it by n -> D^a mod n
    - Alice sends bob 8 and bob send alice 7
    - Alice takes 7 and raises it to her secret number = 7^3 mod 11 = 2
    - Bob takes 8 raises it to his secret number 8^7 mod 11 = 2
    - They compute same secret number
7. Bob does the same with the number he receives from Alice -> C^b mod n
    - These numbers will be the same! I.e. **D^a mod n = C^b mod n**
    - Thus they have a shared secret on a public channel that Eve is not aware of without brute force
    - This protocol has a nice property -> forward secrecy : the past exchanges cannot be compromised

    - We have this property because we use session keys (One for every session) and the session key is thrown away after every session !
- No authentication but we have confidentiality
- Pros
    - Forward secrecy - if keys are compromised in the future who has recorded the conversation - even if they record the conversation and even in the future if they crack the stuff they still can't crack the old ones
    - If you are engaging with someone - u used DH to get a shared key and you decrypt message with the key and then delete the key
    - Even if everything is recorded there's nothing that tells you what the key is
    - Even in the future breaks in its no help to them because you never recorded the secret key
- Weaknesses of DH
    - There are algorithms that solve discrete log - but still alot of work
    - Faster methods than brute force however not much faster
    - ssBaby step, giant step, index calculus algorithm , number field sieve etc
    - *denti*
        - Allows for pre-computation attack i.e. using lookup tables (trade space for time) like logjam attack when common moduli are in use
        - NSA has apparently cracked most modern cryptography yikers
    - Quantum Algorithms! They can be used to break a lot of hardness based protocols
        - Shor's Algorithm
    - Man in the middle attacks
        - Alice would send her messages to Eve and establish a shared secret with Eve and Eve could do this same with Bob
        - Deep packet inspection -> done by large companies/ corporations to watch the packets sent to see if there is disloyal communication -> done through dodgy certificates
        - Having backdoors and making protocols weaker like this to allow 'the good guys' to get in also allows 'the bad guys' to get in :(

Week 8 Foundations- Monday 5th April 2021

Podcast Episode 8.4: Accidents, Safety Culture, Just Culture

The 1986 'Challenger' Space Shuttle disaster

- A disastrous launch led to the tragic deaths of 7 astronauts.
- Disaster was caused by the O ring seal that held the liquid fuel containers together getting compromised.
- The O ring issue was caused by the extremely cold weather on the day (−13 °C).
    - The O ring did not expand
- 2 types of fuel - liquid & solid
    - Chose to use liquid fuel cause cheaper
    - Fuel had to be stored in large fuel containers
    - Constructed in different state and assembled on site
    - O ring used to stop leakage between different sections

The Rogers Commission Report

- The Rogers Commission Report was a special report conducted to address investigate the disaster and discern what exactly went wrong
- One of the members of this commission was Richard Feynman, who soon after the initial report was released, penned and released a dissenting report with different findings
- When Feinmen asked the Senior employees of NASA what they believed the chance of a disaster occuring to be, they reported an expected failure rate of 1/100,000. However, when Feinman asked lower level engineers (the people actually working on the shuttles) the figure they gave was close to 1/100.
- On the day of the accident, *multiple* lower level employees warned that the conditions were unfit for a launch. Despite this, the outside pressures (e.g. the immense amount of news coverage) led to the launch occurring anyway.

Diane Vauhn - The Challenger Launch Decision

- Diane's book, the challenger launch decision, examines the series of events that lead to NASA launching the rocket in such cold conditions
- When initially investigating the incident Vauhn believed the cause was going to be a mix of these three reasons:
    - production pressures
    - risk taking
    - regulatory failure
- In actuality, they found the true causes of the incident to be:
    - A dangerous organisational culture (entrenched secrecy, culture implicitly encouraged employees to keep their mistakes to themselves)
    - The "Normalisation of deviant behaviour"
    - A lack of understanding on how to change the culture once the issues were brought to their attention.

Wedging

- A term coined by our lord and saviour, Donald Trump
- Can be defined as the common practice of a task being assigned to an individual or team, that task having constraints placed upon them, and the inevitable lessening of the task's quality due to the constraints.
- For example, a team of software developers are tasked with creating a website for a company with a nice and responsive user interface. They also are given a deadline for when the task needs to be complete by. The team works on the project.
  The deadline has nearly arrived, and the website isn't fully fleshed out. Because of this, the project lead makes the decision that they can't include *all* the features they really should include in the final product. They decide that the security of the site is the lowest priority, so the website ships with a huge security vulnerability.
- Wedging is borderline inevitable with all projects that involve constraints.

Just Culture

- Many corporate environments have a culture of 'Last Touch Punishment', wherein the last person to sign off or 'touch' the project, is going to be the one to receive all the blame/punishment if a flaw is found within.
- A ***just culture*** is one that tries to take a contrary approach to 'Last Touch Punishment' culture. In a just culture, the focus is shifted towards finding the root

cause of an error/disaster if one occurs, as opposed to trying to scramble to find a scapegoat for the incident.
- A just culture ideally praises people who point out errors. This incentive means mistakes are more likely to be found and are more easily rectified before they spiral into a larger issue.

Week 9 Core Monday 12th April

Richard got a haircut! His t-shirt also sports the graphic "Evil genius", he would make a good villain.

Today's book: The Right Stuff by Tom Wolfe

Magic Trick (no trick involved) just pure magic
3 cards, randomly pick one, depending on which card was picked, Richard showed a different location which would predict that that card would get picked.

- Ace of diamonds was written on the back
- Seven of clubs was written in the "wand" used to choose the card
- Queen of hearts was written on paper inside the card pack

Checks

- Tension between engineering security roles and GRC security roles
- GRC = Governance, risk management, and compliance
- Engineering people do technical
- GRC do checking and monitoring and making you do forms and comply with government
    - They are annoying to engineering people
- Engineers scornful about compliance? Surely not
    - OAuth 2.0 exists????
- Reasons we need GRC roles:
    - Same reason you have checklists of things you have to do
- Richard brings in interesting book
    - Talks about the Black Box. If plane blows up then we learn what went wrong so we can fix it in the next one
    - Talks about the recordings of when a crisis happens
        - The people start going through a checklist of what might be the problem
        - This is what Richard loves about checklists - When you are stressed you can rely on a checklist.
    - Test airplanes are always about to crash so people are always paying attention

    - Well defined sequence of people to ask (bubble up consulting, there will always be someone who will know)
    - If something is critical, don't leave everything to the last minute as it causes errors in critical thinking
    - Richard always has a packing list on his fridge so that whenever he needs to go somewhere he can just check his packing list so he won't forget anything
- We shouldn't be scornful about GRC
    - They are actually very cool #GRCGANG
    - Checklists help you not forget things
    - Only thing bad is if they're too onerous- you spend all your time on it

Richard time travels 2 weeks ago and goes on a tangent about Apollo 13

- Recommends watching interview with the engineer that refused to sign off on the launch
- Engineer says biggest regret that he didn't push back harder
  Richard goes on another tangent about the SS California (tangent - or effective communication through storytelling? :))
- Shot up a flare but only 1 ship saw it
- Flare was thought to have been from the Titanic so captain of the other ship didn't take action
- Luckily the SS Carpatheon came to the rescue
- For every 20 minutes in the water, half the people died (people have a half life of 20 mins in freezing water)

Security by Design

- Instead of adding at the end, it should be built-in from the very start
- Don't leave the security people off the table
- Hasn't happened much in the past because
    - Systems were already built
    - It wasn't viewed as being very important
    - The people who were in charge were old
- Richard praises Sigmund Freud for 3 mins straight
    - Psychology
    - Some ideas a bit mental
    - Psycho-analysis
    - Old guy at conference always brings up Freud
        - Smart young minds are quiet because they don't wanna fight and speak up
    - **The field advances one funeral at a time - need to get rid of old ideas**
- Security by design is now possible because cooler engineers now (gets a seat at the table)
- Steering columns in cars used to kill people "anti safety spear"- cause they were like spears
    - What are air bags called? Richard does not know but if someone finds out please tell him

    - Sharp safety spear
    - Ford did analysis to find out if lawsuits cheaper than improving safety of car and decided cheaper to pay lawsuits
- Privacy people aren't invited to the table yet - from Richard "sucked in privacy people"
    - They are what security people used to be - "privacy by design" seen as costly and can be shoehorned as an afterthought at the end
    - We must take care of them instead
    - "Privacy gets 2 people, security gets 100 people"
    - Privacy people get - Broken chair, secretary 2 days a week and sits in the basement
        - It's a hard life for privacy people
- Why should we implement security by design? Some security measures are hard or impossible to do when security is added in afterwards instead of throughout
    - By including security by design at the table, it lifts the profile of security
- When things are fundamental to the design, you can't just nail them to the outside. Have to design them in when you are designing
- People that influence have the power
    - Soft power
    - Speaking up at the table, arguing for security
    - Changes direction of the whole ship

Richard suggests following these steps for solving things (even in the exam). Like a checklist :)

- Checklist - Richard doesn't pack the morning he leaves :)
    - Helps you work out things
    - Helps you stop overlooking something obvious
    - Know the checklist!
- Research what you need for a trip to thailand - Richard suggests searching for "What do I need for a trip to thailand"
    - He plagiarises other people's checklists
- Checklist should be structured
    - Into categories if using a top down design
    - Despite Richard's fondness of using a checklist with priority queue categories, he still runs around last minute packing
        - Don't rule anything out - don't say "Oh I don't need to take any wet weather gear to Thailand", there can be some good ideas
- Edward de bono approach to problem solving
    - Fairly unstructured approach and try not to be critical - not trying to prioritise
    - And then try to group and prioritise but try not to do premature optimisation as it constrains thinking
- Simulate going to Thailand
    - Simulate ideas that we haven't even thought about

- Recovery strategy
- Don't do it yourself - Have many people around you, the more eyes you have the better you can see the problems (design teams should be run by spiders)
    - The more diverse the better, so you create more ideas.
- Keep thinking about it everyday and tease out ideas
- "How do we know when enough stress testing has been done? Because the process keeps going on"
    - "Yes." - Richard
    - Up to you as a professional, Richard doesn't have a proper answer.
- ISM standard by the ASD, the ultimate checklist (but you can't just have one checklist and say you're good)
- Since technology and things are constantly changing you have to constantly go back and check thing
    - The point is not to get a tick or get a certificate, we're trying to be secure
    - Our job is never finished, there'll be more security things in future
    - But there is no limit because you will always be defending in a changing environment
- Richard used to be an Actuary - his first career
    - You can have a boring life or an awesome life
    - Parents only thinking about type 1 error
        - What if you survive but your life sucks?
    - Tech + Law = jobs for the rest of your life!!!

====== Break time :) Please take 5 minutes from reading these notes ======

"When life gets tough, get a haircut" - Richard

You need different ideas

- Have a million ideas and then cut them down

Voices at table aren't diverse = shitty decisions

Can't get too comfortable- harvest the best idea, don't have a single point of failure

Checklist 2.0??

1. Identify the assets (what we're trying to protect) → don't overlook an asset (don't defend the wrong thing)
    - Usually we put all our efforts into one or two things and we forget about the other things
    - Don't want you to notice an asset after its gone
2. Threat Modelling → Identify the possible attacks and sources
    - Richard's cool hacker friend says skip this step - Richard thinks he's wrong

        i. Sharpens your focus when doing threat modelling
    - What are the sources of attack that you need to worry about?
    - Imagine scenarios
    - Think about what level of access and resources each attacker has.
3. Looks at the risks/vulns
    - Risk registers, keep for compliance for auditors to let them know you have security brains
        i. Identify risk register
            1. Something you show to auditors to show that you've thought about the risk
            2. What measures you've put in to reduce the chance of it happening, reduce impact if it does happen, detect if it does happen

- Ask yourself in the event of a data breach, if the information leaked compromises existing security.

Story time with Richard - Needle mugging story

- Walking home late a night, grabbed from behind, syringe to his neck
- Threatened with aids from blood filled syringe
- Richard went super saiyan (dm me if you can imagine this)
- Went in to get an aids test → would have to wait 6 months for results → right when they wanted to have kids
- Used bayes theorem to calculate if he had aids
    - Richard: “what is the error rate on the AIDS test”
    - Doc: “5%”
    - Richard: “TYPE 1 OR TYPE 2?????”
      Doc: “shut up pls”
- Richard’s friend in surry hills (back when it wasn’t gentrified) left fancy car window open so people can break into it without breaking the window (cheaper to get burgled than to replace window)
    - Loses a couple bucks every week instead of replacing window

Magic Trick 2.0

Richard where is my magic trick
It’s called a magic trick because we have been tricked
HERE IT COMES (it never came)

Week 9 Core Tuesday 13th April

Why Communication?

1. **Cybersecurity is a sudden, new risk** – risk that we are not familiar with, not trained before, not in the education system or discussed about.
    - In comparison to crocodiles and cars: you know their behaviour if you are familiar. Just like security, when things first appear, there is no norm, you don't understand it.
    - Attack surface is huge and new – old and young people all using phones and computers without knowledge of cybersecurity
2. Everyone in the community should receive education about security. People need to respond to current situations (instead of knowing them and still no action).
3. Inward skills (problem solving etc) vs Outward skills (communication): As Uni students, practice communication skills, so later you can make an impact on organisational decisions.
4. Depth vs breadth: have a few things you are good at, and don't have skill set gaps.
    a. Don't just be deep

Subway Story

- Everyones walking around on platform
- Suddenly someone falls
- People rush to edge to see what happens then look to side and see train approaching
- People waving to train driver to STOP STOP
- One person jumps down on track and saves person
- Train flys past

- Moral: You've actually got to do something if you want change
    - Don't be the person just posting and leaving comments - this shouldn't make you feel like you're doing something
    - Actually do something!

Train video?
https://www.dailytelegraph.com.au/news/national/man-saved-after-falling-onto-sydney-train-tracks/video/4827ffaf25b40bedd7769ef25bbddd05

How to communicate

Powerpoint

- Why are they boring?
    - Reading slides
    - 500 words = distraction

Good communication at a party

- React positively
- Putting their own ideas forward
- Contributing back
- Listening

- US Navy calls powerpoints - hypnotising chickens

Communication happens when both parties pay attention and put in effort. Good conversationalists make others good conversationalists (feedback).

Leaflet story: Richard used to hand out leaflets door to door. Once an angry man(a politician) stormed out and threw Richard's leaflet away and yelled "don't give me this rubbish". Richard did the opposite, he tried to put all his leaflets in that man's letterbox. The guy was rude and didn't communicate well. If he asked nicely Richard would've stopped giving him leaflets.

Why was the man (aka little Hitler) angry? He wanted to bring about a change/achieve his purpose. But really wanting change does not bring it about. Communication without purpose is just a puff of wind.

- Maybe his dad died from papercut
- Maybe he's an environmentalist
- Final answer: he wanted to bring about change
- His objective was to have a change in him

    - Stupid - it should have been to bring about a change in me (Richard)

To bring about change, you need to know your audience

- Effective communication is about them not you

Step 1 of communication is deciding on your objective.

Department of Finance recruitment ad: "look at how great we are"

Communication tips(high level)

Think of the change you wish to bring about to your audience, and everything you do revolves around that. Example: The goal of a wedding speech is to convince everyone that this wedding is a good idea.

Communication top down vs bottom up

Tell people what your objective is at the beginning of your talk, don't talk about all the pieces and finally form the big picture, people will get confused in the middle.

Communication tips(Concrete)

1. Use narratives
    a. We remember stories more than events
2. Gain your audience's attention: change of topic, change of voice, give breaks, use activities(but with a purpose cuz no time to waste during speech)
3. Don't use acronyms.
    a. This is just a power thing! Don't do it!!
4. Test your communication plans.
5. Involves the audience in creation.
6. End with a call to action. Get people to do something that achieves the change you want to bring about.
7. Use empathy:if you want to convince the government to save refugees, tell real refugee stories, let politicians feel how refugees feel.

Which movie are we watching today? -> The China Syndrome (he mentioned last night)

Week 10 Engineering - Monday - done by F13B

None of the guest speaker content is assessable? (It is though, iirc.)

Guest speaker **Professor Monica Whitty** - Specialises in psychology and security

- Why human factors relevant to cyber:
    - About sociology, business models
    - Tech needs to be driven by understanding of humans: cater to human needs and capabilities
- Scams
    - Looking out for bad spelling and grammar - but everyone makes spelling mistakes and scammers can write good english

Why is psychology important

- Online relationships
    - Understanding and fighting cyber romance scams (learning what to detect)
        - What do people lie about
        - Who is likely to fall for scams
        - When looking for a romantic partner, do you look/search for criminality? (probably not, this is why they are effective scams)
        - Cognitive dissonance they really know it isn't real but they convince themselves - relates to confirmation bias
    - Gender differences
        - In a test, when men do well, they attribute it to how smart they are but when they do not do well, they blame it on the test
        - Women tend to be the opposite
    - Trend
        - Those who scored higher on having idealised romantic beliefs were more likely to be scammed
        - Similar with investment scams
- People who have been scammed once, are more susceptible to being scammed again.
    - They could get put on a “suckers list”.
    - “Suckers lists” are very valuable.
    - Algorithms are being developed to detect scams

- There is a lack of sympathy for victims
    - Rape, domestic violence
    - Society thinks its their fault for being scammed
- There is a scam out there for everyone.
    - Victim blaming is not a good policy.
    - Shame prevents reporting incidents and getting help
    - Training package for the policy about this topic
    - Police don't really understand and advise them to just not do it again
- Scams tend to cause more damage to society compared to the benefit gained by the scammer. What is lost means more to the victim compared to the scammer. Suicides that result can also have a ripple effect on others being impacted.
- Shame is also a big part of it
    - Romance scam
    - Attack on trust
    - If you click on a malicious link, it's your fault
- Human factor is important
    - What is being attacked is the things in the middle
    - Social engineering, written by humans
    - To be effective and bring about change, its stems for human
    - Human person needs to understand cyber and it's about communicating it better
- Covid app
    - If they got the medical experts, instead of government to introduce the covid app, it might've worked better
- Importance of education
    -
- PURPOSE: improve policy, detection algos
- Engineering bridges theory and practice, psychology bridges humans and security
- People who go on the government's anti-scamming website are more likely to get scammed!
    - After visiting they think they know everything they need to know.
    - Govt doesn't do a great job at explaining the scamming methods

Communication and **education** is important to fix issues

- Examples of this working are : skin cancer, STI
- This is what needs to happen for security

Ppl who fall for scams are similar to those in cults

- They believe things and only see what confirms their belief
- Cults tire you out - make sure you only focus on their beliefs; application of consistency
    - Scammers apply this technique

Need to fix the problem at the root

- Scamming isn't seen as bad as stealing so it's not as big a deal - root problem lies in the community and culture
- Criminals learn from each other and test out theories - have a sharing community

---

Normal Lec Content Resumes:

LAST MONDAY LECTURE POG (not pog I'll miss RB lectures :( )

Security patterns

- Think of things not as list but as a pattern i.e. chess, music
    - Describing it from a high level
- Good thing about patterns
    - 1. Can use it as a checklist to go through and think about patterns you haven't thought about
    - 2. Patterns tend to have common things in them, so you can identify more patterns
- Security pattern examples
    - Security by obscurity
    - Reliance on secrecy - gives us false sense of confidence
    - T1/T2 errors
    - Fail visible, fail safe
        - We want to design things in a way that we are able to see things visibility when they fail and when they fail, it's not a catastrophe
        - Failure critical component is a component that if it fails, the whole system fails - we want to build something with no failure critical component so you design it in a way that when it fails, there's a backup
        - **Fail safe:** In engineering, a fail-safe is a design feature or practice that in the event of a specific type of failure, inherently responds in a way that will cause minimal or no harm to other equipment, to the environment or to people. (wikipedia)
    - Deny by default - don't allow things unless there's a reason to allow it
        - Not only will have you defended against the weaknesses, but you are also defending against the new ones that people haven't thought about yet because by default, they will be denied as well

- Allow lists/deny lists (formerly whitelist and blacklist) - security should be set up with allow lists
    - Having an allow list is better than having a deny list
    - Eg. list of softwares that you can have on your machine and you can't install more unless its on that list
- Principle of least privilege - giving everyone the least privilege they need in order to do what they need to do
    - Don't give them super user access by default
- defence in depth
    - Triangulating in on it
    - Two walls instead of one
    - Don't just rely on your outer wall to defend your castle, also have an inner wall
    - Avoid single point of failure if you have back up plans
- separation of control
    - Whenever you have unreliable components in a system which might fail,
    - People - get two people to launch
- cohesion / coupling / complexity - focus on decoupling to avoid domino of failure, increase cohesion, reduce complexity
    - Control your errors
    - Managing errors
    - Exposed to malicious adversary
    - Increase Coherent/cohesion - everything should just be doing one job, not two
    - Reduce Coupling - a failure in one spot shouldn't lead to a failure in another spot ; minimize coupling
    - Reduce Complexity - weeds where the bad guys will hide
- watch your trust
    - Don't just trust everything
    - Ask yourself hard questions
- think like an attacker
- Wedging - difficulty to change once you've committed to something; wedged between constraints
    - e.g. specific due date so people might cut back on security
    - Committing to something in the future meaning you will be subject to all these constraints and everything you're depending on all start changing but you cannot change the final date
- Use well known processes
    - Every new thing is another risk
- Standards - standards self improve

- herd immunity and collaboration - learn from each other
    - Everyone works together in a security community
    - Any new piece of knowledge can be quickly shared to everyone else and that immunizes us all against that particular attack
    - Rising tides lifts all the boats
- Community
    - Eg. in medicine
- think like an attacker
- Checklists
    - Important so that you don't forget things
    - Note binding are great so that you don't forget things and are organised
- test and attack
    - Test and attack everything
    - Build something and try prove that its wrong
- Skepticism
    - Be skeptical of things
- measure, quantify - help us analyse
- control impact not just likelihood
    - not m&m, bulkheads on ships
    - Think about it like - if it did happen, how can i limit the impact of it happening
- review, many eyes, diversity
    - Have other people reviewing what we do
    - Help pick up things we are blind to seeing
    - Tip: tell someone else before you transfer money
- Allow for humans
    - align incentives - align someone's self interest with what you want them to do
    - accountability, conflicts - people who aren't being checked/watch
    - cognitive vulns
    -
- monitor, close the loop - put a check in place to check the system
    - Close the loop: continuously get feedback
    - Don't do things that aren't being checked
    - Put a check in place to check the system
    - Close the loop = get feedback
- Never be sure of anything, never done, never move on
    - Mindset that nothing is ever perfect
- delete delete delete - value forward secrecy, water
    - Delete data that you don't need so it doesn't come back to bite you later on

- Tranquility - assuming people don't move around e.g. even after people leave they still have access to the database
    - Assumption of tranquility is wrong
    - Proving something is secure is dangerous because how can you know assumptions are true in the real world.
    - Changing roles
    - They did not relinquish his access
- don't mix data and control
- training - like public health
- influence up and down
    - How to influence people above and below you
    - Keep them educated
- professionalism, integrity, values
    - To society
- overconfidence, humility, hubris
    - Be humble

Week 10 Core – Tuesday

Root cause analysis

- what caused people to make these dumb decisions?
    - Keep asking why
- What is wrong with system that lead to this behaviour?
- E.g. if you replaced the CEO would they make the same mistake? The issue could be that higher ups don't listen to what engineers say, break down of communications, no reward for reporting up, Conflict where the person who is trying to save money is the same person who is trying to fix problems or perhaps penalising people who highlight problems.
    - Solution might be to have anonymous reporting mechanisms
    - Solution might be to have higher ups do compulsory work on the 'front line' with engineers eg. Every board has to have at least one engineer on it
- Life protip: written complaints generally have more follow up than verbal
- A bad workman blames his tools
    - Similarly a CEO would want to blame individual people

Simplification is great, but to fix problems we can't afford to simplify too much - blaming someone Is oversimplifying the problem

- R. Buckland recommends reading history
- We can take a step back and see where the future is headed
- History repeats itself - sometimes we can see where the rollercoaster is going to lead (ie. the expected outcome)
    - Tranquility problem with retired politicians - they still have influence over certain people or domains bc they become lobbyists
    - Politicians are lobbying his proteges or supporting people they supported before
- Tranquility - where you assume that in a static frozen position, everyone has roles and responsibilities that balance each other and everything looks fine but that was based on a flawed assumption that any of the rules would never change to another role

Commitment:

- Commitment replaces Trust.
    - If two parties are trying to decide on something remotely, need some way of both parties making a commitment that cannot be changed and use that to decide
        - This removes the need for a third party to replace the trust between the two parties
- How can you toss a coin and know that the outcome is fair when the two people are not in the same room
    - Trying to replace trust in someone with something that is beyond our control But something that is also inevitable Ie there is onewayness
- Paula's Idea:
    - Paula chooses heads or tails and write it on a piece of paper (ie. you have to committed to it), and concatenate their answer with a random number of your choice (nonce)
    - Hash it, and make the hash public and that your public commitment
    - You announce if coin is heads or tails
    - The other person can't change their answer because the hash proves their original choice
- Richard's card magic trick from last week
    - He didn't commit to how he was going to reveal what he 'chose' in advance
    - **If commitments aren't made public, then they are no good**
- Commitment removes the need for trust

- Final approach
    - I will suggest a word or number
    - You will take my number and stick heads or tails after it and then stick your number to the end and hash it
    - Because i've had a contribution to it, one person does not have control of what she is hashing

Exam:

- 3 different sorts of qns, with different weights
- But sorts of qns and amount of writing is similar
- Some technical, analysis, problems solving, creativity, case study, predict future, tracking current news, qns about activities from course
- Biggest thing is ANALYSIS, so analytic responses
- Moodle NOT inspera!
- Will get a encrypted text file of exam in case moodle doesn't work (can get password from course staff?)
- Skeleton of exam will be provided (layout with the qn content removed, but mark weights kept)
- Will be the same paper for 6841/6441, but you stop at different points depending on

- which course you are doing
- The exam is held at 13:00-17:00
- The movie is Apollo 13 (watch it if you haven't!!)
-

Exam Topics (lets make notes on these):

- I'll try linking the relevant parts for these topics and add extra notes/Q&A for these - feel free to contribute as well
- RSA SSL
    - Calculations
    - Bits of info
    - Bits of security
    - Can use wolfram alpha
    - Law ppl disregard technical stuff **and just imagine what's gonna be in the exam and what's not lol good luck**
- Cracking classical codes
    - One time pads
    - Substitution ciphers
    - Transposition ciphers
    - Vigenere ciphers
    - Any other cipher variant
- Risk
    - Type I type II
- TLS / SSL / internet security protocol
    - How authentication works
    - Authentication strengths and weaknesses
- **Enigma machine** (the paper thing we did a while ago)
    - Should have the enigma machine ready to go!
    - **PRINT IT OUT and have pencil and paper ready to go**
- Binary file editor (know how to use one)
- Estimation
- Communication
    - Analyse strengths and weaknesses of a piece of communication
    - Creation of communication
- Hashes, MAC, HMAC
- Blockchain
- SHA, OpenSSL, Other things in the open SSL suite - web ones are okay, but you can use ones on your machine if you want (see the birthday attack hints page:

https://www.openlearning.com/unswcourses/courses/sec-21t1/activities/birthday/hints/?cl=1)

- Using a hex editor
    - For the hashing exercises we use it to validate and end of line characters
- **Watch Apollo 13**
- Root cause analysis
- Just culture
- Normal accidents
- Privacy and data
- Communication (as in how to communicate effectively)
- GDPR (Europe), FOI (Commonwealth), gipper
- We won't be asked stuff like what are the 5 steps of TLS connection handshake
    - But might be asked "an attacker is using TLS to attack a website, how might they be doing it?"
- Richard can't think of how to put ethics in the exam, so probs are not going to be asked..
- What is analysis exactly?
    - Looking at things from different perspectives could be an example of analysis
    - Analysis is an approach
    - I could tell you the wavelength of blue length that your eyes pick up
    - Why do you think the primates developed red and green receptives
        - Why it happened
        - When it happened
    - Not repeating facts but taking it and seeing a bigger picture

Richard's Summary of course:

- Engineer solves problems not by magic but a framework built over many years and past failures
- Cyber Sec Engineers have yet to build a perfectly secure system- why?
    - Cyber Sec isn't really a discipline or perfect science yet
    - So we learn from mistakes, case studies etc
- Risk is invisible, we should make it visible as much as possible
- When analysing problems, always get diverse views
- Analyse where you are and where you are going in life
- Work with others as much as possible
    - Diversity

Put clocks back in the room when you are late…. Lmao don’t actually

Notes on Exam Key Topics:

(Thank you to everyone who contributed. You guys rock!)

Apollo 13 Main Points

Characters in Apollo 13 (2013)

Astronauts

Jim Lovell | Tom Hanks

Fred Haise | Bill Paxton

Jack Swigert | Kevin Bacon

Ken Mattingly | Gary Sinise

Ground Crew

Gene Kranz | Ed Harris

John Aaron (EECOM) | Loren Dean

Sy Liebergot (EECOM) | Clint Howard

List of Important Characters in the movie (Only the important ones, add more if required)

- mentions apollo 1 practice launch that killed its passengers
    - Jim Lovelle/Tom Hanks said that they fixed the problems with it (door)

    - His wife has a nightmare about the mission going wrong
- The team is exposed to measles (due to another employee who just tested positive and was in contact with the 3). Two of the crew have already had the disease and are hence immune, but Ken has not and the incubation period is longer than the time before lift off, so they cannot predict if he has actually caught it.
    - This breaks up the crew and they have to go with someone else as substitute
    - The well oiled team that has been training for ages suddenly gets change 2 days before the lift off
- Simulations with substitute doesn't go well
- Engine 5 dies - as long as they burn the others should be ok
- Fred starts throwing up
- Jack stirs oxygen
    - Something goes wrong - wires come loose?
    - Multiple failures occur - at least 4
        - Immediately think it's an instrumental failure rather than the slim chance of 4 things all failing
        - Contacts mission control for advice, tells them to stay calm as they think of a solution
        - Suddenly, things get worse, and mission control are now in a frenzy to find a solution to fix all the problems (electrical, oxygen, etc.)
    - Losing oxygen (venting)
- Systematically going through what could have gone wrong
- Turn off power and close off fuel
    - kills mission to the moon
    - saves data and does manual calculations - verified by several different people
- Earth crew were arguing about what to do
    - Least problematic option but takes too long vs
    - Other solutions that are unexplored but will get them to earth faster
- Talking to literally everyone involved in the building of the shuttle
    - Trying to save as much power as possible, since power is the only thing keeping everything else running (computers, oxygen)
- Getting earth crew astronauts to do landing simulations to be able to figure out how to do the reentry
    - Ken Mattingly called in to do landing simulations due to his experience with the ship
- Create carbon dioxide filter using random stuff on board
    - They are first making it on earth and detailing everything they do
    - Then they recall it to the astronauts to copy the procedure
- Never know what sequence of events will occur that will lead to a situation - Jim
    - when he was talking about algae leading him home

    - but he could only see the algae when his lights died
- Added extra sticky notes over buttons that would detach, leaving behind the other astronauts
    - double safety
- This mission was a successful failure

Solving "Bits of Work" Questions

- Average == divide AMOUNT OF WORK by 2 (i.e. minus 1 bit)
    - NOT the number of bits (divide BITS by 2 for birthday attack)
- TOTAL == take the amount of bits/work directly

- 10^3 = 1000 = 2^10 (just assume that for these questions)

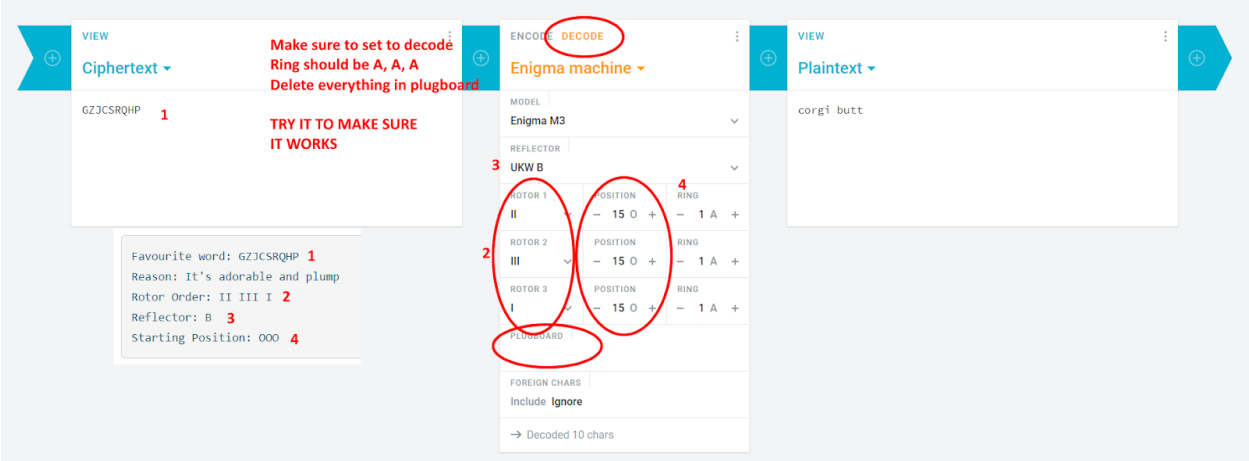
1. Write out all the **options** you have for a given problem
    a. e.g. "given a password with x y z constraints"
        i. how many x or y or z's are there?
    b. "given a picture of a person's face"
        i. how many eye/nose/ear/face shape/etc types are there? (HELP
2. Add them all together to get number of options
3. Check if you have multiples of the options
    a. e.g. 6 character password means the number of options multiplies 6 times
        i. (x+y+z) * (x+y+z) *(x+y+z) *(x+y+z) *(x+y+z) *(x+y+z)
        ii. or (x+y+z) ^6
    b. One person's face means we don't need to multiply - we just keep the number of inputs we have
4. If it asks for the
    a. Total amount continue
    b. Average - divide amount of work by 2 (subtract 1 from the power)
    c. Any match/collision (birthday attack) - sqrt (divide power by 2)
5. **TO GET THE NUMBER OF BITS** - convert it to a power of 2 and the power (exponent) is the number of bits of work

Questions:
Question 3: Suppose I have a password that is 6 characters…
Question 5:

Enigma Machine - how to do it online

- online enigma machine.
- https://i.imgur.com/
- This website has an sII0gi0.png bigger image of below
- (https://www.101computing.net/enigma-machine-emulator/ also has an Enigma machine. You can see the rotors turn, and it gives you detailed output if you want it (what mapped to what). It also looks suitably 1930s :p It's slower than the above cryptii one though, since it's letter by letter.)

Maybe i am a bit late. But what does the reflector B do in Richard's example in activity?
(To above: B is the default reflector in the Mark 3 Enigma, the one that we used in flat Enigma activity, so we choose this in online emulators as well. A reflector maps one letter to another, and then passes the signal back through the rotors so it goes through another round of scrambling, it then comes out as the final output.)



-

Authentication

Authentication

- **Authentication Factors:**
  - What you are = a fingerprint
  - What you know = secret
  - What you have = to verify a physical attribute like a fingerprint, you may require a transducer
  - What you can do = e.g. a skill

- Once they find the secret they could crack everything, only needed one factor to crack all factors, knew every secure factor making it no longer secure
    - All authentication is really just one factor
- Two factor authentication on phone: what you know and what you have (your phone), send a SMS to your phone to authenticate
    - Someone could forge a sim card - that's just a secret
- Authentication: daily, visible action completed all the time, logging into computer, phones, checking into work etc .
    - Visible, measurable, tangible action

AUTHENTICATION PROTOCOLS - Fall Back on when confused

- Authentication products - lots of jargon, "gobbledygook"
    - vendors "sending dreams", probably only caters to one error
    - Offering a miracle for one error but a nightmare for the other
- There'll always be **type 1 / type 2 errors** in authentication
    - E.g. getting money out of bank account, two errors: bad guy getting money out of my account OR I can't get money out of my account
    - When t1 errors go up, t2 errors will go down and vice versa. Finding balance is difficult
- Authentication ties a request to an **identity (and this is a DIGITAL identity)**
- **IDENTITY** never going to be an actual real world identity - a piece of text (phone number, email address, TFN)
    - Can only tie a request to a piece of data, can never truly authenticate that it's your mum talking to you
    - "They have access to the phone number"
    - Phone/email not a terrible proxy for identity - you have usually have one phone, check your email regularly and keep it close
    - BUT email address can be owned by multiple people, read by vendor/supplier, company email, shared password etc
- Visibility of authentication has 3 properties
    - Annoying - pressure to make authentication less annoying
    - Attract funding - "space age, scifi" "more advanced and agile than others - fingerprint scans, facial scans, biometrics" justify big spends. People like spending on visible things than intangible things
    - When things are visible, and fail in the visible way - pretty good for security BUT bad guy always try to do authentication silently

- Authentication vs authorisation
    - Authentication = linking identity in a certain way / Make sure you are what you are
    - Authorization = giving permission to undertake an action
    - Usually authorise yourself by logging in (get a **token** to do things as whoever you've logged in as e.g. GitHub personal access tokens), token authorises you to keep having access to a resource
- Authentication - timing ,duration, frequency
    - Authenticate constantly (eg. each keystroke) -very annoying
    - Usually authenticate less frequently and trusted between times
    - Authenticate at the beginning and think it's ok then on
        - Have authentication tokens at critical times but most of the time just have authorisation to allow you to do anything for a while
            - Privileged activities might need more authentication but most activities just rely on the initial authentication of logging in
    - M&M property - protect the border - assume everything bad is on the outside
        - Temporal M&M, outside thick side is authentication then on inside it's just authentication yum chocolate soft

    - **Authentication depends on Parties involved**:
        - Look at all these different authentications that have their own challenges and protocols:
            - client authenticating server (e.g. you logging into gmail)
            - server authenticating a client (site telling browser that it really is amazon.com)
            - peer authenticating other peer
        - A danger of authentication: sometimes it's easy to forget that authentication is like a web spun between all different parties involved. Don't focus on just one connection between 2 members and think that's all the authentication that is needed
        - Most of the time, there are multiple parties involved in a system and they need authentication to one another, however we have the tendency to only focus on authenticating 1 or 2 parties.
    - **CHECKLIST ^**
    - **Penn and Teller episode**

Authentication Protocols

- Challenge that you will reveal information by authentication but want to prevent replay attacks from occurring
- Only way to authenticate in most cases is to convince we know a secret
- Challenge response: mum what did i eat for dinner last night, answer changes all the time, preventing replay attacks
- Proof of liveness: check that the person is actually there

  E.g. banks need to use a card to log in, get into building, go pee → need to take it everywhere, proof of liveness as if the card is there you probs are too
- **S/Key**
  - Use the rightmost key first (server needs to remember this key), then the key just before it (server can verify hash of key is the last key), and so on, each time server just needs to remember the last key and check that the hash of the each key offered matches the most recently used key.

    Uses hashes
    Had to type password in the clear if sniffing
    Password that can only be used once (one time password)
    Start off with an initial secret (could be your own password to log in) → hash it with a good hashing algorithm e.g. SHA3 → take hash and hash it again with SHA3 write it down → hash again → hash again
    hash(hash(Hash(Hash(Hash…..))))Keep hashing the hashes
    - Have a list of all hashes down but the computer system only lets you to use them all one time
    - Computer can check that hash is correct instantly, but hacker cannot go backwards with a hash
    - Can be used a finite number of times, finite, but clever
- TOTP: temporal one time password

  Time orientated, doesn't use counter for no. of times used but a counter that updates every 30s
  How can time-based authentication be exploited?
  - Attackers can brute force to race you to gain access if they see the first few characters of your verification code
  - If the website accepts multiple authentications then they can get the same token as you if they take your code and verify within the same time-frame

- HOTP: hashed one time password
  Have a counter → every time you use it counter increases by 1 → get counter and hashes using HMAC (keyed MAC, you and server knows the shared secret)
  Allows for infinite number of times to use password unlike S key
  Google authenticator uses this
- Can use RSA backwards, encrypt a message using private key and then send it to someone who can decrypt it with my public key → good way to authenticate i.e. see if you know your private key

Signatures

- Both physical and digital signature should ensure authentication and integrity
- Signature should authenticate you as you and ensure the integrity of the document (that it's been read and agreed to and it won't be changed after you sign it)
- You can use RSA to do ^:
  - Using RSA: Can encrypt the entire document using private key (creating a ciphertext) and then attach it to the document, and that would be a signature!
  - To verify: decrypted the ciphertext using public key and it should return the same document the ciphertext is attached to. This works because only you know your private key. Provides authentication and integrity of document.
  - Non-repudiation: You can back out of any contract by saying "oh no i lost my private key! Any hacker could have used my private key!" and now all the documents containing your signature is rubbish :)
  - Problem 1: The person writing the contract has control over the wording. You don't want to use your private key to encrypt something other people have control over like that because they could be planning an attack (Someone could reverse and find your private key). General rule: don't sign document other people have produced.
    - Prevention: we usually hash the document before the encryption (signing).
  - Problem 2: What if the document is bigger than one block in the RSA (RSA only does a block of a certain size). Now you have problem with block modes
    - Prevention: we usually hash the document because it makes it a fixed size

- OAUTH

    Single sign on which spans across different organisations i.e. would you like to sign in using google account?
    First authenticates by logging into google then it sends authorisation tokens to other websites to allow you to do things using the google account
    Google, facebook, amazon, microsoft, twitter and, and, and, and
    Convenient, annoyance massively reduced but what are the trade offs?

Authentication Attacks

- Fallback attack:
    - If you fail to authenticate too many times, server will degrade to lower authentication standard (e.g. WIFI uses WPA if you do not have WPA2). Bad guy can force the fallback to be used.
- Password recovery:
    - Attack the password recovery workflow (social engineering, weak fallback questions whose answers can be recovered via social media)
    - Set up a email made ***only*** for password recovery, make it really strong key, don't use for anything else, if someone compromises this then attacker has everything
- SIM cards:
    - Reddit attack in 2018: 2FA over SMS, bad guy bribed AT&T employee for $75 to gain access to reddit employees' SMS deets
- Interfaces and transducers:
    - Hardware written by third party, trust foundation is wider. Insiders and backdoor. Risk minefield, higher chance of failure.
- Session Hijacking:
    - Steal your authorisation token
    - Shown in the youtube video in lecture, (https://www.youtube.com/watch?v=xaOX8DS-Cto)
- No Lock-Out
    - No limit on how many times you can enter an authorisation token so you can just brute-force until success

Cracking Codes

Cipher	Encrypt	Decrypt	Cracking
Substitution cipher			
Caesar cipher	Shift the alphabet and encrypt/decrypt corresponding letters	Shift the alphabet and encrypt/decrypt corresponding letters	• Brute force • Analyse letter frequency

Vigenere cipher	Each letter is coded based on a key word	Each letter is coded based on a key word	• Find the key length using
			  ◦ Kasiski's method
			  ◦ Index of coincidence (https://asecuritysite.com//encryption/ic)
			  ▪
			• Break the ciphertext into n length chunks
			  ◦ https://www.quora.com/How-can-I-crack-the-Vigenere-cipher-without-knowing-the-key
			• We now have n Caesar ciphertexts
			• Analyse letter frequency along columns and map to correct letter
			  ◦ https://www.guballa.de/vigenere-solver

Permutations cipher	Break into groups and jumble the letters	Break into the length n and read columns	• Find the key length
			• Break into key length
			• Shuffle columns until key length makes sense
			  ○
			  ○ https://www.dcode.fr/transposition-cipher

Techniques

Order Of Frequency Of Single Letters	E T A O I N S H R D L U
Order Of Frequency Of Digraphs	th er on an re he in ed nd ha at en es of or nt ea ti to it st io le is ou ar as de rt ve
Order Of Frequency Of Trigraphs	the and tha ent ion tio for nde has nce edt tis oft sth men
Order Of Frequency Of Most Common Doubles	ss ee tt ff ll mm oo
Order Of Frequency Of Initial Letters	T O A W B C D S F M R H I Y E G L N P U J K
Order Of Frequency Of Final Letters	E S T D N R Y F L O G H A K M P U W
One-Letter Words	a, I.
Most Frequent Two-Letter Words	of, to, in, it, is, be, as, at, so, we, he, by, or, on, do, if, me, my, up, an, go, no, us, am
Most Frequent Three-Letter Words	the, and, for, are, but, not, you, all, any, can, had, her, was, one, our, out, day, get, has, him, his, how, man, new, now, old, see, two, way, who, boy, did, its, let, put, say, she, too, use
Most Frequent Four-Letter Words	that, with, have, this, will, your, from, they, know, want, been, good, much, some, time

-

A MISPLACED DECIMAL POINT WILL ALWAYS END UP
N JAVULNEHT THEAJNL UZAWM GALL NLGNSV HWT BU

WHERE IT WILL DO THE GREATEST DAMAGE. A
GDHOH AM GALL TZ MDH YOHNMHVM TNJNYH. N

NARROW MIND HAS A BROAD TONGUE.
WNOOZG JAWT DNV N QOZNT MZWYBH.

Cipher solved! You are awesome! You solved the cipher:
A MISPLACED DECIMAL POINT WILL ALWAYS END UP WHERE IT WILL DO THE GREATEST DAMAGE. A NARROW MIND HAS A BROAD TONGUE.

-

Clear | Enable Jump | Change Cipher

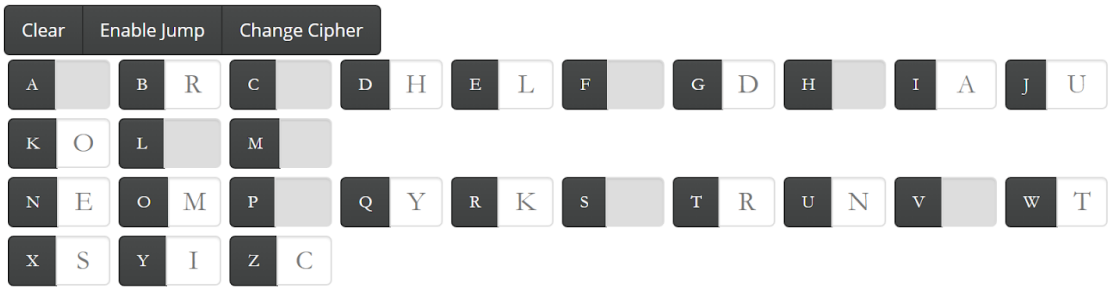
A	I	B	U	C		D	H	E	C	F		G	W	H	E	I		J	M
K		L	L	M	T														
N	A	O	R	P		Q	B	R		S	Y	T	D	U	P	V	S	W	N
X		Y	G	Z	O														

    - Look for patterns and common word pairings
    - Look at common doubles
    - Try to crack one word (has, where)
    - Think about it grammatically
    - When you have a pattern (such as -arrow) or missing one letter, brute force it

- Try it anyway

Welcome to the **UNSW SECedu Cyber Foundations** NSA Game Challenge Bot

```
IN SOME COUNTRIES, CHAUCER AND DANTE ARETHE CLASSICS.
YU XKON ZKJUWBYNX, ZDIJZNB IUG GIUWN IBNWDN ZEIXXYZX.

        IN THIS COUNTRY, IT'S A SORT DRINK.
        YU WDYX ZKJUWBQ, YW'X I XKTW GBYUR.
```



-
  - The apostrophe told me it was in S
  - I found the common two, three, four letter words first
  - The word country had the same prefix and different suffix, meant it was the same word
  - I brute force the C and found the word CLASSIC
  - The type was the ARETHE

DOES ANYONE KNOW WHERE YOU CAN GET A BLANK VERSION OF THIS WEBSITE APP THING WHERE YOU CAN FILL IN YOUR OWN CIPHER AND SOLVE IT?

https://www.boxentriq.com/code-breaking/cryptogram this has one but it also has an autosolve thing.

Type I and Type II Errors

Question: Question 2: You are assessing an email spam filter. If…
Type I and Type II Error (This is a common theme in Security problems)

Test say yes and you have it	Test say no but you do have it (false negative = type II)
Test say yes but you don't have it (false positive = type I)	Test say no and you don't have it

Improving a Type I or Type II error is a Zero Sum game: one will improve, another will suffer.

A type I error is the rejection of a true null hypothesis (also known as a "false positive" finding or conclusion; example: "an innocent person is convicted"), while a type II error is the non-rejection of a false null hypothesis (also known as a "false negative" finding or conclusion; example: "a guilty person is not convicted").

Note that which is type 1 error and which is type 2 depends upon initial assumptions of the problem- the table above implicitly assumes everyone does NOT have COVID. If we instead assume that everyone does have COVID, the error types switch around.

** I feel like row1 col1 and row2 col2 should swap if it's ppl NOT have COVID case.**

I.e. Improving welfare

Type I issue: The most disadvantaged cannot access welfare payments

Type II issue: Some are taking advantage of the welfare payments when they don't need it.

Type I solution: Make welfare more accessible (Disadvantage: Type II suffers; more ineligible people access welfare)

Type II solution: Make welfare less accessible (Disadvantage: Type I suffers; more ineligible people will not gain access welfare)

Fixing a security error may decrease the chance of that error occurring, however, it may increase chances of another error as a by-product.

- Improvements will usually err on the side of the **visible problem**
- Improving on the **invisible problem** can cause annoyance

TLS, HTTPs, SSL

Podcast Episode 8.1: Encryption on the web - TLS, SSL, HTTPS

Old days on the web

- Very limited encryption
- Everything sent in plain text
- Had to assume no sniffing
- Local area networks setup so anyone could see everything
- Use telnet to login remotely (all passwords sent in the clear)
- Everything was in the clear
- Mosaic browser was graphical
- People could see what you're looking at
- Internet was a delivery service for information
- Interaction with the webpage was also in the clear
- If you wanted encryption, you had to do it yourself
- As serious systems went online - realisation that we need encryption
- Netscape - early browser company
    - Had a famous cryptographic hero
    - Secured info
    - SSL would replace Berkley socket
    - Abstraction
    - Secure socket layer (SSL)
    - HTTP over SSL = HTTPS
        - Independent of user
- First version of SSL in early 90's
- SSL 2.0 1995
- SSL 3.0 1996

- **Can trick server to use weaker security**
    - Mr nah from picking Netflix movies
    - Fallback attacks deez nuts
- Wasn't a priority to upgrade to better SSL
- NHS used to use Windows XP for way too long
- Microsoft didn't see internet would be a big thing
    - Introduced internet explorer later on
    - Broke the standard because other browsers didn't support certain things
        - Deliberate (as a strategy)
    - Microsoft just wanted money
- Changed SSL to TLS (Transport Layer Security)
    - Just so it didn't look like a netscape thing
- TLS 1.0 1999, TLS 1.1 2005ish & stayed around until recently (2020ish)
    - Turned out to be vulnerable as well
    - Accepted use of protocols that were broken (RC4, MD5 etc.)
    - Stupid cheap browser
    - Fallback attack
- TLS 1.2 2008
- TLS 1.3 2018
    - Everyone should be using this now
    - Became very tough
    - Throws out lots of algorithms that are weak and compromised
- Used the intel holographic sticker to advertise on computers

How does HTTPS negotiate and set up connection

Properties

3. TLS ClientHello contains Server Name Indication (SNI) / handshake phase
    - Required to serve multiple HTTPS sites from the same IP
    - Things u want / properties (CIA properties):
        - C - Confidential (actual data should be invisible)
            - Ciphers
            - Encrypted
                - The actual data is encrypted (the rest of the URL)
            - Visible / not encrypted
                - IP address

                - Port number
                - Volume and timing
                - Who you're talking to - e.g. google.com
            - Symmetric cipher (Stream or block cipher)
        - I - Integrity (don't want people messing with data)
            - Series of blocks sent in this connection over these sockets
            - End of block - message authentication code attached
            - All block ciphers in TLS 1.3 - ciphering mode does integrity checking
            - Don't want people to fiddle with what we're saying
        - A - Authentication
            - Forward secrecy
            - Browser authenticate that its talking to the right server
4. Communicating
    - Use cryptography

Hand shake (5 Steps)

6. Client connects to server. (client doesn't know what protocols server can and can't support)
7. Server responds to say which protocols/cryptographic primitives will be used (normally one encryption and one hash to do authentication and confidentiality)
8. Server then sends a certificate to the client and this certificate authenticates the server through a process called pki
9. Client checks that certificate is valid
10. Client generates a temporary session key that the client and server will use to

- In security don't trust anyone or anything
- TLS encryption uses a symmetric cipher
- TLS 1.3 is savage as it doesn't accept anything even potentially insecure, e.g. cipher block chaining as there was an attack discovered 'lucky 13'
    - Only mode uses is a variance of counter mode, GCM, CCM

Understand these:

- Fallback attacks

    - Tricks computers to abandon newer more secure versions of operations to older less secure ones
- Poodle attacks (SSL 3.0)
- Heartbleed
- Don't ever roll your own crypto

Attacks

- **Poodle attack (Padding Oracle On Downgraded Legacy Encryption)**
    - **Overview**
        - Is a vulnerability common to SSL v3.0
        - Is Man in the middle + fall back attack
        - whereby you force a session to fall back to ssl v3.0 and then go to town
    - **How does it fall back?**
        - You just interrupt the client to server handshake enough times that it forces the session to connect vis ssl v3.0
    - **Then?**
        - if cipher suits RC4 or block cipher in CBC mode
        - Attacker can retrieve partial bytes of encrypt text, and later get full plain text
    - **References**
        - https://www.alertlogic.com/blog/poodle-the-man-in-the-middle-attack-on-sslv3-d62/
        - https://www.youtube.com/watch?v=XxO7L5gHshY
- **Fallback attack**
    - man in the middle attack
    - interrupts connections and thus pushing systems to use legacy cryptographic protocols
    - which are then susceptible to hacking
    -
- **Heartbleed attacks**
    - **Overview**
        - When a client and a server communicate with one another they connect via SSL
        - Servers tend to host a lot of sockets, so it terminates some of these sockets
        - **To prevent the server from closing their socket**, the client sends a message (saying hey im still here) known as a **heartbeat**

        - This data is not encrypted.
        - **Furthermore, these heartbeats can be of varying sizes less than 64kb**
        - And servers respond in kind with the same size eg.
            - client sends 1, server sends 1
            - client sends 64, server sends 64
        - eg. heartbeat actual packet size = 1kb, but says it is 64kb,
        - server responds with 1kb and 63kb of random memory
        - could hold passwords, usernames anything
        - these can be sent in quick succession and numerous times
        - Getting useful information
        - The computer that received the heartbeat request never checked to make sure the request was actually as long as it claimed to be. So if a request said it was 40 KB long but was actually only 20 KB, the receiving computer would set aside 40 KB of memory buffer, then store the 20 KB it actually received, then send back that 20 KB *plus* whatever happened to be in the next 20 KB of memory. That extra 20 KB of data is information that the attacker has now extracted from the web server.
        - This is the crucial part of the operation. Even when a computer is done with information, it persists in memory buffers until something else comes along to overwrite it. Attackers can take that extra 20kb memory and potentially find crucial info.
        -
    - **References**
        - https://www.youtube.com/watch?v=6Sz5wBBXzpc
        - https://www.youtube.com/watch?v=WgrBrPW_Zn4
        - https://www.csoonline.com/article/3223203/what-is-the-heartbleed-bug-how-does-it-work-and-how-was-it-fixed.html
        -

Episode 8.2: PKI, X.509 and the Ministry of Silly Walks

- Go amazon ip
    - How do we actually know this is amazon?
- Use secrets
- Knowing a secret no one else knows could be enough
- 2 different approaches to do this on internet
- Approach 1: Web of trust
    - Crowdsourced truth
    - Talk to enough people that are trusted that can authenticate
    - Decentralised system

    - Used by famous email encryption (PGB)
        - Everyone exchanges keys when they meet
        - Key signing parties
        - Accumulate a whole lot of signatures
    - Problems
        - 1
        - 2
            - Certificate authorities can sign certificates from other companies and create a link of certificates who trust one another that may be malicious
- Approach 2: Pyramidical Structure
    - PKI - Public Key Infrastructure
        - Solves the key distribution problem
        - Anyone can see your public key
        - Use your private key to decrypt
        - But how do I find your public key?
            - On webpage - which webpage do I trust?
            - Which key belongs to which person?
            - Man in middle attack
            - Can promote false keys and trick people!
        - Central system tells everyone what the public key is
            - Insert false key in central system?
        - When you ship browser - **browser has certificates preloaded of which authorities to trust**
            - Use to be able to just pay to be included as a preloaded certificate
        - Are certificate authorities trustworthy?
        - **Ties domain name to certificates instead of companies**
- It's hard to revoke certificates!Called X.509 certificates because they used to be part of a wider protocol
    - Protocol was never set up to enable easy revokal
    - Used to use revocation lists but no actual mechanism to create or distribute them (especially now because they can be huge and there would be so many of them)
    -

Episode 8.3: DH, Forward Secrecy and MITM F

- Diffie Hellman Key Exchange

    - Raise 2 to a power and then mod it
    - This problem is known as the discrete log problem
    - DH relies on the one-way hardness of this problem -> can't get much faster than brute forcing with this problem
    - We have Alice and Bob who want to communicate securely but they never got together to establish a key
    - On a **public channel** they agree on what number they are modding by a prime (n) and a number that they are raising to a power (g)
    - They each think of a secret number a and b (must be between 0 and n-1)
    - Alice does a secret computation and computes C = g^amodn
    - Bob does the same computation for b -> D = g^bmodn
    - Now they exchange their answers -> Eve would have to resort to brute force to calculate a and b
    - Alice takes the number that Bob sent to her and raises it to her secret power and mods it by n -> D^amodn
    - Bob does the same with the number he receives from Alice -> C^bmodn
    - These numbers will be the same! I.e. **D^amodn = C^bmodn**
    - Thus they have a shared secret on a public channel that Eve is not aware of without brute force
    - This protocol has a nice property -> forward secrecy : the past exchanges cannot be compromised
    - We have this property because we use session keys (One for every session) and the session key is thrown away after every session !
- Weaknesses of DH
    - Faster methods than brute force however not much faster
    - Baby step, giant step, index calculus algorithm , number field sieve etc
    - *denti*
        - Allows for pre-computation attack i.e. using lookup tables (trade space for time) like logjam attack when common moduli are in use
        - NSA has apparently cracked most modern cryptography yikers
    - Quantum Algorithms! They can be used to break a lot of hardness based protocols
        - Shor's Algorithm
        - 😊 check out my Something Awesome https://www.openlearning.com/u/krittika/blog/QuantumInformationTheoryAndTheEvolutionOfCyberSecurity/ if you wanna learn more :)
    - We can hear Richard's dog snoring :)
    - Man in the middle attacks

- Alice would send her messages to Eve and establish a shared secret with Eve and Eve could do this same with Bob
- Deep packet inspection -> done by large companies/ corporations to watch the packets sent to see if there is disloyal communication -> done through bodgy certificates
- Having backdoors and making protocols weaker like this to allow ‘the good guys’ to get in also allows ‘the bad guys’ to get in :(

Using Hex Editor

- Using a hex editor
    - For the hashing exercises we use it to validate and end of line characters
    - https://hexed.it/

Birthday Attack hints and resources

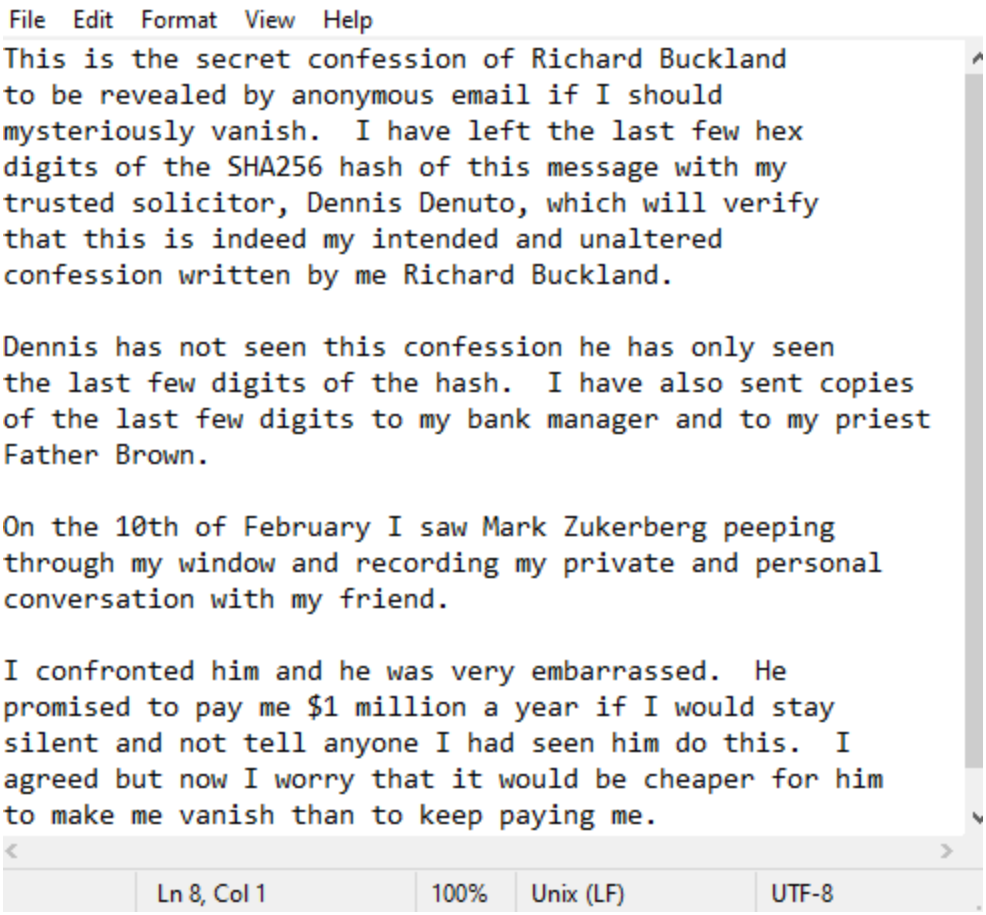
How to alter text files so they look the same, but are actually different (ie so they generate different SHA256 hashes but look the same to the eye?)
The confession files have more than 10 lines each. One simple method is as follows: If a line ends in a space call it 1, if a line has no space call it 0. Count up in binary to generate different combinations of spaces and no spaces. That will generate 2^10 combinations (just using the first 10 lines).
eg if lines 1, 3, and 5 have a space at the end, and all the other lines don't, then that is variant number 0000010101

How to compute SHA256 hashes fast?
Coding/Scripting:
If you are a programmer and want to code it by calling a crypto library (or just want to script openssl) that's fine. In that case see how many digits of the two SHA256 hashes you can get to match. The tips below are for everyone else.
Manually:
You can do it manually on a website like

- https://hackerstoolkit.net/sha256-checksum/.
- This site computes SHA256 hashes for files, AND
- you can also enter and hash text directly, which is convenient for manually trying lots of variants fast.

Text File format
Windows sometimes uses two special characters CR-LF to mark the end of lines of text files. Linus and mac tend to just use LF. (LF is the ASCII code for Line Feed it is code OA in hex, CR is the ASCII code for Carriage Return, it is 0D in hex).

All the interactive online SHA256 generators that I've tested so far (including the one listed above which I recommend as being quite useful) assume that when you are typing text into the web page and press enter to make a new line, they assume that the file being hashed has a LF inserted at that spot.
So if you are a windows user who is generating and testing the confession variants using an interactive online SHA256 generator - then when you have found matching confessions make sure you save them using a text editor which uses linux/unix style end of line characters, not windows CR-LF pairs, or your file's SHA256 hash won't match what you saw in the interactive window. (Notepad for example will list what line endings it is using if you are on windows 10)



Also make sure your text file, and your testing text, has a new line at the end of the last line.
If you want to see what is going on view your text files using a hex editor. it should end in a solo LF (ie 0A) , and not a CR-LF pair (ie 0D0A).
Below are two sample files just containing the word "testing" you can down load as tests. One terminates the line using LF, one uses CR-LF. You can inspect them using a

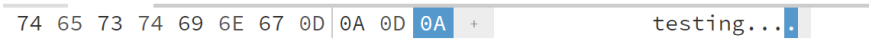
hex editor to see the difference. Their corresponding SHA256 hashes are also shown. Type testing into an online SHA256 generator and check that it generates the correct hash. Then create a text file yourself containing just the word "testing" (without the quotes) and terminating with a LF and SHA256 hash that to test that you are creating text files correctly.

- **testing-lf.txt**: (unix/linux/mac format) - SHA256 hash ends in "...c2"
- **testing-crlf.txt**: (windows format) - SHA256 hash ends in "...2b"

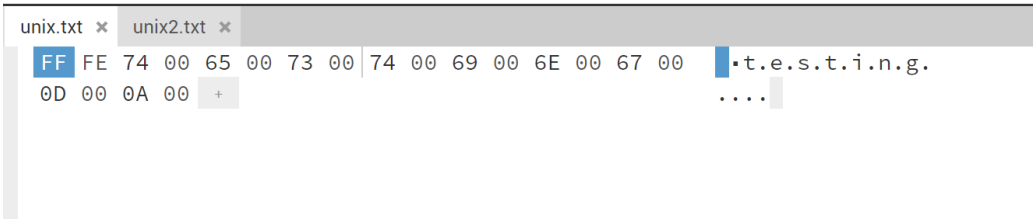
From <https://www.openlearning.com/unswcourses/courses/sec-21t1/activities/birthday/hints/?cl=1>

- Testing end of files
  - Used https://hexed.it/
  - File 1
    - This file contained the word "testing" and no space

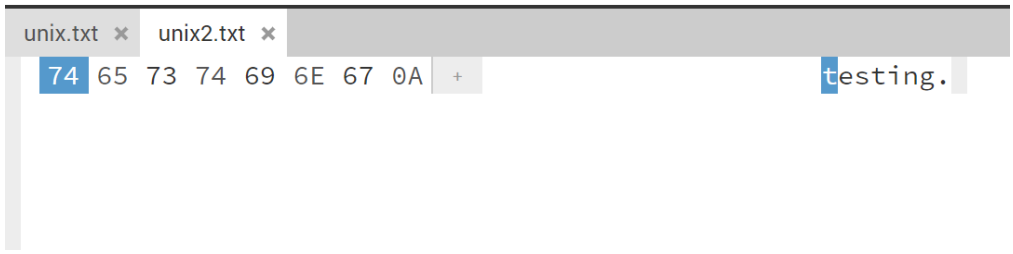
  - File 2
    - The file contained the word "testing" followed by 2 newlines



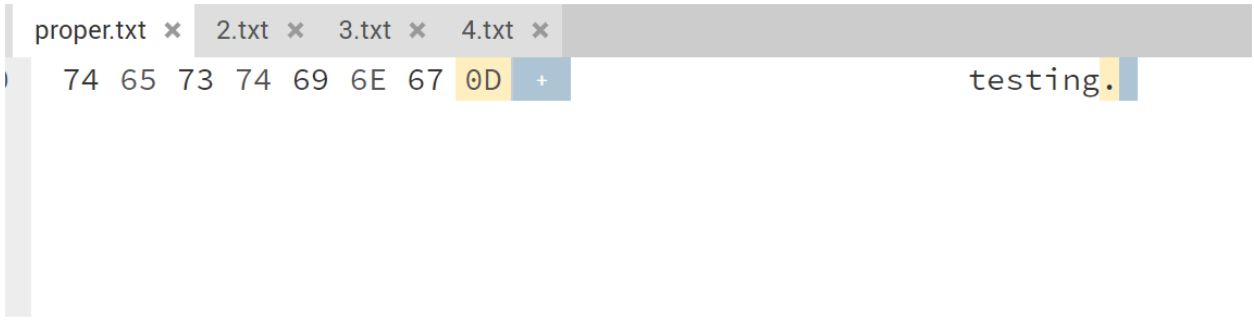
  - File 3
    - Created on Windows Powershell



  - File 4
    - Created on Ubuntu



- Solo LF = 0D = carriage return **this is not true. Solo LF = 0A = line feed**
- CR-LF pair = 0D0A = carriage return and control character = newline
- Make sure that the confession files end in the same thing by opening them in the hex editor



Estimation

Reasonableness Check

1. How big is an ant, how many ants would make a green pea (say 5?)
2. How many peas fit into some other object of slightly larger size (e.g. whiteboard eraser)
3. How many of the whiteboard erasers fill a section of the room (something easy to estimate within reason)
4. Repeat increasing the size of the next step up until you get to the size of the room
5. With each step, choose a size that you are confident in the estimation of the number of previous sized objects fit inside of

Is It Reasonable?

Claim 1

57% of the world's population lives in the Northern Hemisphere

Implausible - there is more than that

Claim 2

Jeff Bezos makes (on average) $321 million per day

Implausible - less than that

https://usa.inquirer.net/64935/how-much-does-jeff-bezos-make-a-second Pretty crazy but he apparently does make that much in a day if you just measure his net worth change over a year

Claim 3

Australians buy 250 million tubes of toothpaste per year

Implausible - less than that

Claim 4

One in four UNSW undergraduate students studies engineering

Implausible - there are many faculties

Claim 5

Australia's sheep population is over 50 million

Plausible

Claim 6
On average Americans spend 5475 hours on their phone per annum
Implausible - less than that

Communication and creation of communication

Why Communication?

1. **Cybersecurity is a sudden, new risk** – risk that we are not familiar with, not trained before, not in the education system or discussed about.
    - **In comparison to crocodiles and cars: you know their behaviour if you are familiar. Just like security, when things first appear, there is no norm, you don't understand it.**
    - **Attack surface is huge and new – old and young people all using phones and computers without knowledge of cybersecurity**
1. Everyone in the community should receive education about security. People need to respond to current situations (instead of knowing them and still no action).
2. Inward skills (problem solving etc) vs Outward skills (communication): As Uni students, practice communication skills, so later you can make an impact on organisational decisions.
3. Depth vs breadth: have a few things you are good at, and don't have skill set gaps.
    1. Don't just be deep

Subway Story

- Everyones walking around on platform
- Suddenly someone falls
- People rush to edge to see what happens then look to side and see train approaching
- People waving to train driver to STOP STOP
- One person jumps down on track and saves person
- Train flys past
- Moral: You've actually got to do something if you want change
    - Don't be the person just posting and leaving comments - this shouldn't make you feel like you're doing something
    - Actually do something!

Train video?
https://www.dailytelegraph.com.au/news/national/man-saved-after-falling-onto-sydney-train-tracks/video/4827ffaf25b40bedd7769ef25bbddd05

How to communicate

Powerpoint

- Why are they boring?
    - Reading slides
    - 500 words = distraction

Good communication at a party

- React positively
- Putting their own ideas forward
- Contributing back
- Listening
- US Navy calls powerpoints - hypnotising chickens

Communication happens when both parties pay attention and put in effort. Good conversationalists make others good conversationalists (feedback).

Leaflet story: Richard used to hand out leaflets door to door. Once an angry man(a politician) stormed out and threw Richard's leaflet away and yelled "don't give me this rubbish". Richard did the opposite, he tried to put all his leaflets in that man's letterbox. The guy was rude and didn't communicate well. If he asked nicely Richard would've stopped giving him leaflets.

Why was the man (aka little Hitler) angry? He wanted to bring about a change/achieve his purpose. But really wanting change does not bring it about. Communication without purpose is just a puff of wind.

- Maybe his dad died from papercut
- Maybe he's an environmentalist
- Final answer: he wanted to bring about change
- His objective was to have a change in him
    - Stupid - it should have been to bring about a change in me (Richard)

To bring about change, you need to know your audience

- Effective communication is about them not you

Step 1 of communication is deciding on your objective.

Department of Finance recruitment ad: "look at how great we are"

Communication tips(high level)

Think of the change you wish to bring about to your audience, and everything you do revolves around that. Example: The goal of a wedding speech is to convince everyone that this wedding is a good idea.

Communication top down vs bottom up

Tell people what your objective is at the beginning of your talk, don't talk about all the pieces and finally form the big picture, people will get confused in the middle.

Communication tips(Concrete)

1. Use narratives
    1. We remember stories more than events
2. Gain your audience's attention: change of topic, change of voice, give breaks, use activities(but with a purpose cuz no time to waste during speech)
3. Don't use acronyms.
    1. This is just a power thing! Don't do it!!
4. Test your communication plans.
5. Involves the audience in creation.
6. End with a call to action. Get people to do something that achieves the change you want to bring about.
7. Use empathy:if you want to convince the government to save refugees, tell real refugee stories, let politicians feel how refugees feel.

Which movie are we watching today? -> The China Syndrome (he mentioned last night)

Hashes, MACs and HMACs

HASHES

Good General hash function properties:

- Unique objects may be hashed to the same value (collision). Where objects have different values in attributes that we want to distinguish between, they should end up with different hashes.
- We want even the tiniest change to create a vastly different hash, so we can use the hash function result to detect if something has changed the file or not. It can be a good way of checking the integrity of files cost effectively, given that you have a good hash function
- Bcrypt is the current industry standard for hashing.
- Can be stored regularly without worry of compromising the original file
- You want the things you are trying to distinguish to have different hash values, so collisions only occur between things you don't care about

Good Cryptographic hash function properties:

- 1-way (cannot recover original file given the hash value) AKA **pre-image resistance**
    - No faster way to find file that hashes to a certain value than to try all possible files
    - Works as information is lost through hashing producing a "summary" which can build to multiple things if the method is reversed
    - End result is 'pseudo-random' i.e. end result has no relation to the data it represents, but can be validated when checked that this is the correct input to produce the hash
- Collision resistance:
    - It is computationally infeasible to find two inputs which have the same output.
- Second-preimage resistance:
    - Given one object, cannot find a second object which gives the same hash value as the first object
- Avalanche property:
    - A change in 1 bit of the input should cause EACH hash bits to flip with 50% probability

Examples of Hash Usage:

- Shazam compares a hash of the currently sampled audio across hashes or other such fingerprints in their database.
- Digital Certificates for websites (to confirm a website is legit and not a spoofed site)
- Storing hashed passwords: even if attackers have the hashed passwords, they need to enter the unhashed password in order to produce the hash to log in
    - **CRC (not cryptographic)**
    - **MD5 (cracked in the 90's)**
        - 128bit hash
        - MD2 made in 1989
        - MD4 made in 1990, weaks found in 1991, broken in 1995
        - MD5 made in 1991, weaknesses found in 1993, broken in 2005
        - MD6 made in 2008 (256bit, applied to be SHA-3 and failed lmao)
    - **SHA (NSA made)**
        - SHA-0 made in 1993 (not secure, broken literally right after release RIP)
        - SHA-1 made in 1993 (160 bits, not secure, broken in 2005 ish)
        - SHA-2 made in 2001 (224-512 bits with 64-80 rounds of hashing, sus)
        - SHA-3 made in 2015 (**NIST approved!** 224-512 bits)
            - Different structure to predecessors
        - SHA-kira made in 1977 (Produces absolute bangers, very secure)

"Hash is swiss army knife" - Richard Buckland
Not
Difference between hash and cryptographic hash is that a cryptographic hash should be:
- Preimage resistant (given output, can't work out input)
- collision resistant (given the hash function and you cant find two messages that give the same hash)
- second pre-image 'collision' resistant (given the hash function and one input, you can't find another input that produces the same hashed output)

Ideal Hash Function

- Given a hash(x), it is hard to figure out x from the hash output
- Tiny changes = vastly different hash, to detect if file has changed
- Collision resistant
    - Message authentication code
    - Protects the integrity of message
    - Example: Send A $1 million, banana
    - Attackers can intercept hashed messages
    - Attackers will not know the word "banana", cannot fake pre image messages without the code
    - Can also use a counter (number series so it is unique), stops replay attacks

Man-in-the-middle attacks

Example of the Two Generals problem, hashes are the best thing we have so far to solve this problem even though we know the Two Generals problem technically has no solution

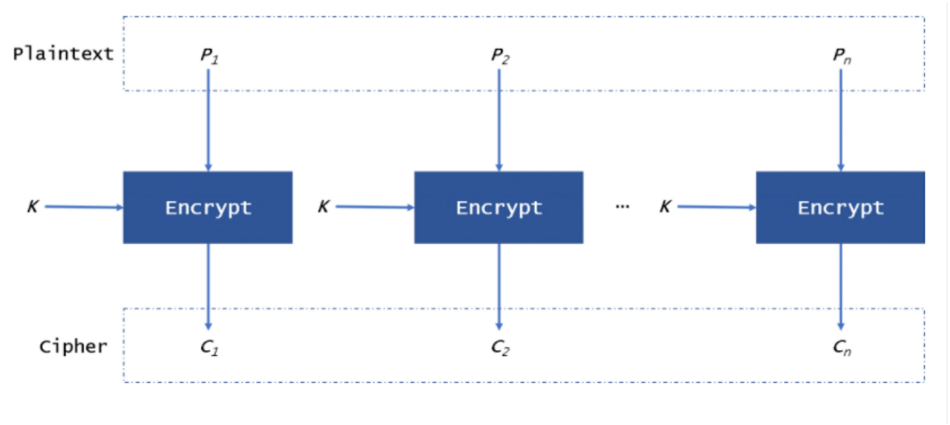
Example of bad hash IRL:

"Here's a video of me hashing a hard drive, and the hash is a perfect record showing that I haven't tampered with the drive at all through the analysis, and that I haven't added or removed files on this drive ;^). I did it with the power of MD2!"

Block Modes

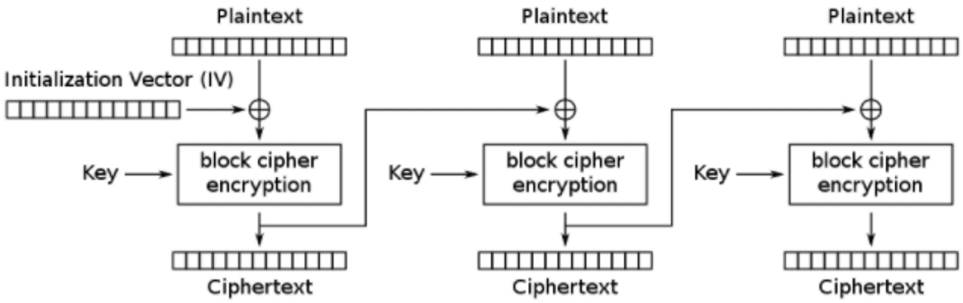
[https://www.highgo.ca/2019/08/09/the-difference-in-five-modes-in-the-aes-encryption-algorithm/]

1. ECB - Electronic Code Book
    1. Vulnerable because blocks can be repeated and it's relatively easy to tell the edges of the blocks
    2. Partition and encrypt each partition, then join partitions together again

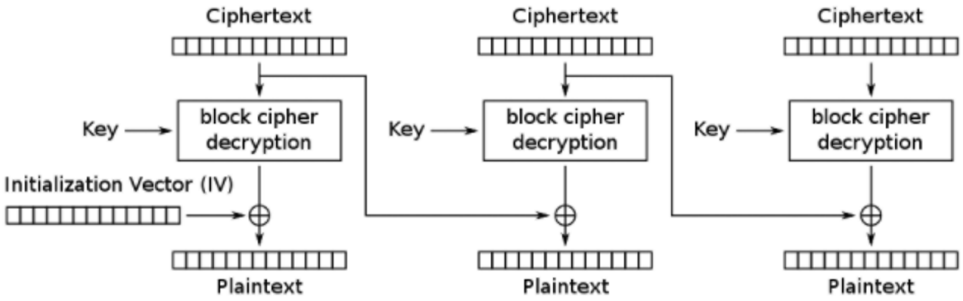


3.

1. CBC - Cipher Block Chaining
    1. Partition plaintext and XOR it with the previous partition's encryption
    2. First partition is XOR'd with some key before being encrypted



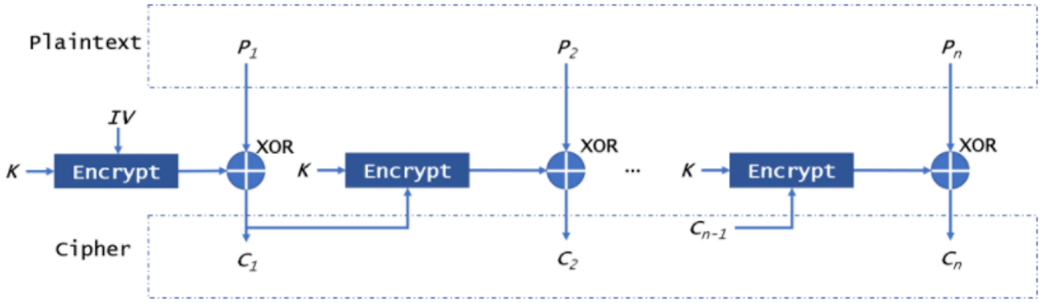
Cipher Block Chaining (CBC) mode encryption



3. BC_encryption.svg Cipher Block Chaining (CBC) mode decryption

1. CTR - Counter
    1. Encrypt the initialisation vector (IV)
    2. Then XOR with plaintext, which makes ciphertext
    3. Repeat

4.


Plaintext
P_1
P_2
P_n
IV
K
Encrypt
XOR
K
Encrypt
XOR
...
K
Encrypt
XOR
C_{n-1}
Cipher
C_1
C_2
C_n

MACs

Description

- MAC = message authentication code
- MAC is a hash function encrypted with a secret key
- Used for message authentication

Table

- Integrity = message has not been modified
- Authentication = message originates from sender
- Non-repudiation = recipient passes the message and proof to third part, confident from sender

Cryptographic primitive Security Goal	Hash	MAC	Digital signature
Integrity	Yes	Yes	Yes
Authentication	No	Yes	Yes
Non-repudiation	No	No	Yes
Kind of keys	none	symmetric keys	asymmetric keys

HMACs

Description

- HMAC = keyed-hash message authentication code
- Specific type of MAC
- HMAC can provide message authentication

Process

- Any cryptographic hash function may be used in calculation of HMAC
- Result is called HMAC-X where X is the hash function used
- The secret key is first used to create two keys - inner and outer
- First pass of algorithm creates internal hash derived from message and inner key
- Second pass of algorithm produces final HMAC from inner hash and outer key
- Does not encrypt the message, message is sent alongside HMAC hash
- Parties with the secret key will hash the message again themselves, and if it is authentic, the hashes will match

This definition is taken from RFC 2104:

$$\mathrm{HMAC}(K, m) = \mathrm{H}\left(\left(K' \oplus opad\right) \parallel \mathrm{H}\left(\left(K' \oplus ipad\right) \parallel m\right)\right)$$

$$K' = \begin{cases} \mathrm{H}(K) & K \text{ is larger than block size} \\ K & \text{otherwise} \end{cases}$$

where

H is a cryptographic hash function

m is the message to be authenticated

K is the secret key

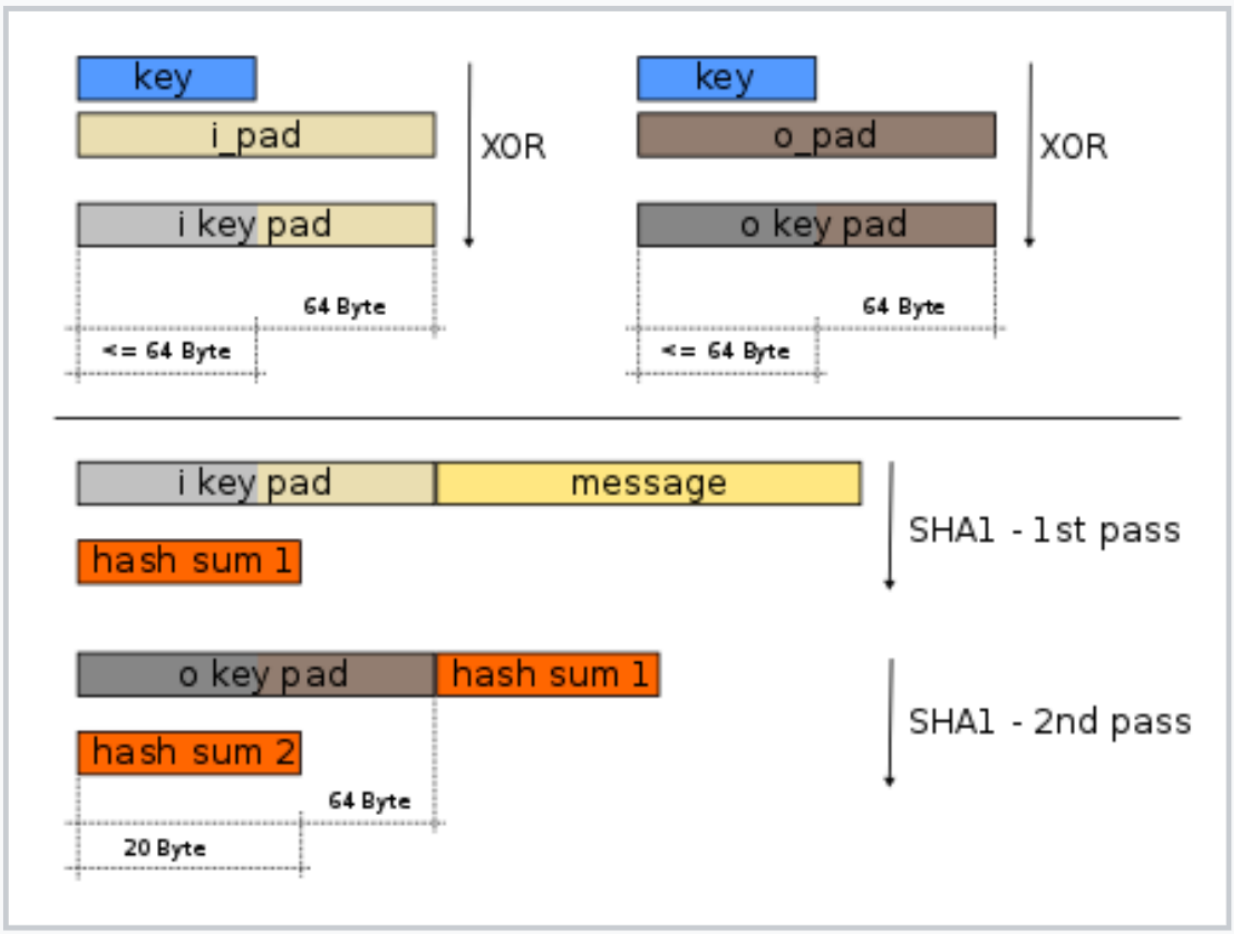
K' is a block-sized key derived from the secret key, *K*; either by padding to the right with 0s up to the block size, or by hashing down to less than or equal to the block size first and then padding to the right with zeros

$\parallel$ denotes concatenation

$\oplus$ denotes bitwise exclusive or (XOR)

opad is the block-sized outer padding, consisting of repeated bytes valued 0x5c

ipad is the block-sized inner padding, consisting of repeated bytes valued 0x36



HMAC-SHA1 generation

Block Chain

Descriptions

- Blockchain is a specific type of database
- They store information in blocks that are chained together

Security and Trust

- New blocks are added to the end, so they are in chronological order
- After a block has been added, it is difficult to go back to alter the contents of blocks unless the majority reached a consensus to do so
- Falsified blockchains will easily be cross-referenced with everyone else's and found to be false
- To succeed would mean the hacker needs to control and alter 51% of all the blocks, but this would be too costly

Hashing a Block

You can use OpenSSL to generate a SHA256 hash for your block. Alternatively you could use an online SHA256 generator like we did in the Birthday Activity. I suggest you make sure you can do it both ways. Below we'll show you how to use openssl.

For example, say your bold data breach prediction is The Tax Office, the previous hash was 0f603b5f322a16568bf7b0acff51008466408cdccbfeff675118bbde8ca49b50 and there is already one block in the chain (i.e. you will be adding the second block in the block chain). Suppose the first serial number we will try will be 0 (and if that does not make a valid block then we'll then just count up 1,2,3... until we find a serial number which makes out block valid)

In our blockchain we will represent each block as a string of text with + signs in between the four fields. So our first attempt at constructing a valid block looks like:
2+The Tax Office+0f603b5f322a16568bf7b0acff51008466408cdccbfeff675118bbde8ca49b50+0

To hash this block you will use the command **echo** to repeat this string on the terminal and use the **|** (äka *pipe*) to send this string into openSSL and to generate the block's hash. This will insert the required LineFeed at the end of the block string automatically.
On the command line this would look like

```
echo '2+The Tax

Office+0f603b5f322a16568bf7b0acff51008466408cdccbfeff675118bbde8ca49b50+0' | openssl

sha256
```

(note we need to wrap our string in quotes)

Try this on your command line, and also using the online generator (press return at the end of the string in the online generator to insert the line feed), and confirm that it gives you the following hash as a result:
4a3e2cf959ebbb280847ab35c1382f583877cfdb8d76d5d593d2c795faf12011
Sadly this starts with "4a" so the serial number is not valid for this block and we can't add it to the blockchain yet. Keep changing the serial number and hashing again, until it meets the BuckCoin valid has criteria (starting in 0 for first hex digit, second hex digit is 0-9)
Tip: You can use the up arrows to access the previous command and use the sideways arrows to go back and forth to change the serial number

The serial number 9 makes our block valid:
$ echo "2+The Tax Office+0f603b5f322a16568bf7b0acff51008466408cdccbfeff675118bbde8ca49b50+9" | openssl sha256
098f43e921f70ec69fd2108a57693bfa1c039267423d06ea812e765dfb7748b8
Provided no one else has beaten you to it in the meantime, you can now go ahead and add your block to the chain. Do this by posting a reply to the comment containing the previous block in the chain (first of all do check that the previous block is itself valid, if the previous block is not valid then your block will not be considered part of the chain ...)
In your comment below include the block string you hashed, and the hash result.
2+The Tax Office+0f603b5f322a16568bf7b0acff51008466408cdccbfeff675118bbde8ca49b50+9
098f43e921f70ec69fd2108a57693bfa1c039267423d06ea812e765dfb7748b8
And you've successfully mined a block! WELL DONE.

From <https://www.openlearning.com/unswcourses/courses/sec-21t1/activities/blockchain/?cl=1>

following Sudaksh:
My Block
Prediction: DELL
Chain Length: 121
Serial Number: 21
Block: 121+DELL+ 0781d888ac8ee99145b54a848fe713d7eb8f31d067f8901f17ced494a37832d3
Hash: 04ffe51276784ef7e986104dfa466bd0c12453f8cfa3ea0a132572eabd28c6be
https://www.openlearning.com/u/shrutikadukar-qoqzia/blog/Week5BitcoinMining/

From <https://www.openlearning.com/unswcourses/courses/sec-21t1/activities/blockchain/?cl=1>

Following Shruti K.

My Block
Prediction: Atlassian
Chain Length: 122
Serial Number: 11
Block: 122+Atlassian+ 04ffe51276784ef7e986104dfa466bd0c12453f8cfa3ea0a132572eabd28c6be+11
Hash: 04b909bfa9da2f4fad89761998a6520cba01d722f6feb5fed44af00e37921a2a
https://www.openlearning.com/u/hanle-qok62e/blog/BitcoinMining/

SHA and OpenSSL

SHA

- Secure Hash Algorithm
- SHA-1, SHA-2, SHA-3
- SHA-1 takes input and produces a 160-bit hash value, NOT CONSIDERED SECURE
- SHA-2, SHA-3 is replacing SHA-1
- Most web browsers are not accepting SHA-1 SSL certificates
- SHA-1 is still secure for HMAC
- CAN BE USED IN PYTHON UNDER HASHLIB LIBRARY

OpenSSL

- Toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL)
- General-purpose cryptography library

TLS Check

- https://hackertarget.com/ssl-check/.

SCAN RESULTS FOR WWW.GOOGLE.COM:443 - 74.125.141.105

---

* TLS 1.0 Cipher suites:

Attempted to connect using 80 cipher suites.

The server accepted the following 5 cipher suites:

TLS_RSA_WITH_AES_256_CBC_SHA	256	
TLS_RSA_WITH_AES_128_CBC_SHA	128	
TLS_RSA_WITH_3DES_EDE_CBC_SHA	168	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	256	ECDH: prime256v1 (256 bits)
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	128	ECDH: prime256v1 (256 bits)

The group of cipher suites supported by the server has the following properties:

Forward Secrecy	OK - Supported
Legacy RC4 Algorithm	OK - Not Supported

The server is configured to prefer the following cipher suite:

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	128	ECDH: prime256v1 (256 bits)

Root Cause Analysis

Root cause analysis

- what caused people to make these dumb decisions?
    - Keep asking why
- What is wrong with system that lead to this behaviour?
- E.g. if you replaced the CEO would they make the same mistake? The issue could be that higher ups don't listen to what engineers say, break down of communications, no reward for reporting up, Conflict where the person who is trying to save money is the same person who is trying to fix problems or perhaps penalising people who highlight problems.
    - Solution might be to have anonymous reporting mechanisms
    - Solution might be to have higher ups do compulsory work on the 'front line' with engineers eg. Every board has to have at least one engineer on it
- Life protip: written complaints generally have more follow up than verbal
- A bad workman blames his tools
    - Similarly a CEO would want to blame individual people

Simplification is great, but to fix problems we can't afford to simplify too much - blaming someone Is oversimplifying the problem

- R. Buckland recommends reading history
- We can take a step back and see where the future is headed
- History repeats itself - sometimes we can see where the rollercoaster is going to lead (ie. the expected outcome)
    - Tranquility problem with retired politicians - they still have influence over certain people or domains bc they become lobbyists
    - Politicians are lobbying his proteges or supporting people they supported before
- Tranquility - where you assume that in a static frozen position, everyone has roles and responsibilities that balance each other and everything looks fine but that was based on a flawed assumption that any of the rules would never change to another role

Commitment:

- Commitment replaces Trust.
    - If two parties are trying to decide on something remotely, need some way of both parties making a commitment that cannot be changed and use that to decide
        - This removes the need for a third party to replace the trust between the two parties

- How can you toss a coin and know that the outcome is fair when the two people are not in the same room
    - Trying to replace trust in someone with something that is beyond our control But something that is also inevitable Ie there is onewayness
- Paula's Idea:
    - Paula chooses heads or tails and write it on a piece of paper (ie. you have to committed to it), and concatenate their answer with a random number of your choice (nonce)
    - Hash it, and make the hash public and that your public commitment
    - You announce if coin is heads or tails
    - The other person can't change their answer because the hash proves their original choice
- Richard's card magic trick from last week
    - He didn't commit to how he was going to reveal what he 'chose' in advance
    - **If commitments aren't made public, then they are no good**
- Commitment removes the need for trust
- Final approach
    - I will suggest a word or number
    - You will take my number and stick heads or tails after it and then stick your number to the end and hash it
    - Because i've had a contribution to it, one person does not have control of what she is hashing

Just Culture

Just Culture

- Many corporate environments have a culture of ‘Last Touch Punishment’, wherein the last person to sign off or ‘touch’ the project, is going to be the one to receive all the blame/punishment if a flaw is found within.
- A ***just culture*** is one that tries to take a contrary approach to ‘Last Touch Punishment’ culture. In a just culture, the focus is shifted towards finding the root cause of an error/disaster if one occurs, as opposed to trying to scramble to find a scapegoat for the incident.
- A just culture ideally praises people who point out errors. This incentive means mistakes are more likely to be found and are more easily rectified before they spiral into a larger issue

Privacy and Data

What Goes Wrong:
All the data that we collect is a huge target, it can be the target for attacks, and if leaked could be used by anyone for whatever purpose. Remember, data doesn't have an agenda, but people do!

Problems with mass data collection

- The purpose the data is collected may not be good (data could be manipulated)
- The purpose the data is collected may not be the only purpose they claim it's used for (possible violation of given consent)
- Data may be misused / leaked by accident / stolen *("Who will watch the watcher?")*
- Type 1 Type 2 Errors

How to detect ID theft:

- Set up notifications for bank activity etc (send alert whenever privileged actions are carried out in your name)
- Request credit history report to detect anomalies
- Request all the information a company has on you to see if someone else is using your identity

Privacy-induced behaviour
Privacy-induced behaviour change depends on how high you are on the primate tree. Higher primates behave more differently depending on whether they are observed. Be monke. Stonke. doge 🐵 💎 🙌. Humans are especially prone to this behaviour, even if there isn't anything particularly sensitive on what you're doing.

A great example: The Panopticon. Panopticons are donut shaped prison design where you're always being watched. Even if you're not being watched, your behaviour will change if you think you *could* be being watched

Governments/Police:

- Stop the spread of covid
- Anti-Terroism

- Taxes
- Census
- Catch criminals and stuff

Companies

- Selling your data
- Outmaneuver their competitors

Our individual motivations

- We want
    - Personalised healthcare
    - Crimes to be caught
    - Democracy
- We don't want
    - Identity fraud
    - Leaked data (from incompetent dataholders)
    - New Zealand secret service /__(👁_👁)__/\
    - Robo debt
        - Literally hounded by the government for a bad system

Steps to being dumb with data:

1. There's something to measure, might as well measure it :shrug:
2. We have all this data, let's use it and analyse it!
3. We've analysed all this data, let's use it!

Forward secrecy/security

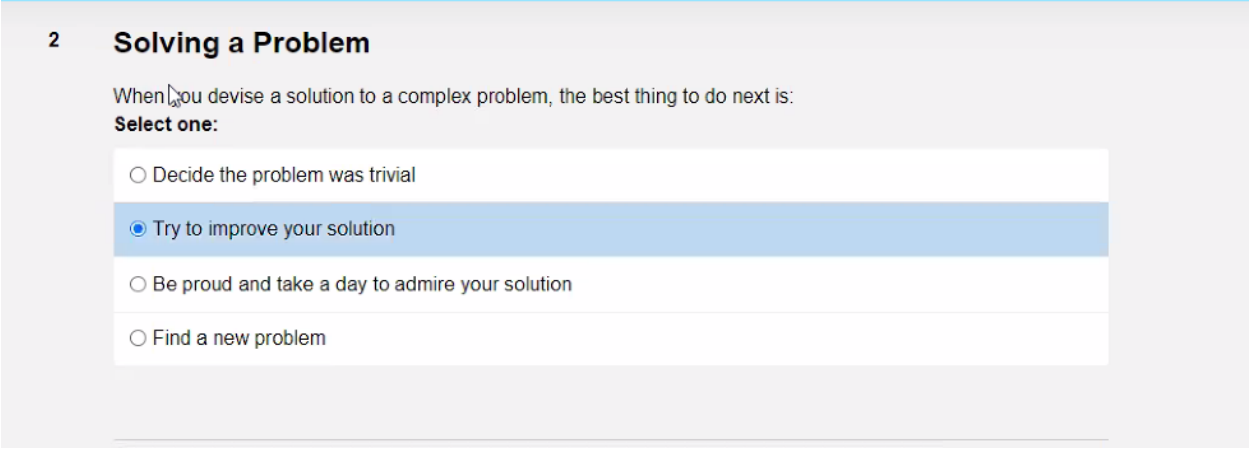
Just because you have nothing to hide *now* doesn't mean you won't have anything to hide in the future. For example: Rwandan genocide, people disclosed their ethinicity in their ID cards; Journalists who reported on the Afghan files may regret providing the government with a long fingerprint on their ordinary behavior, their social connections, etc.

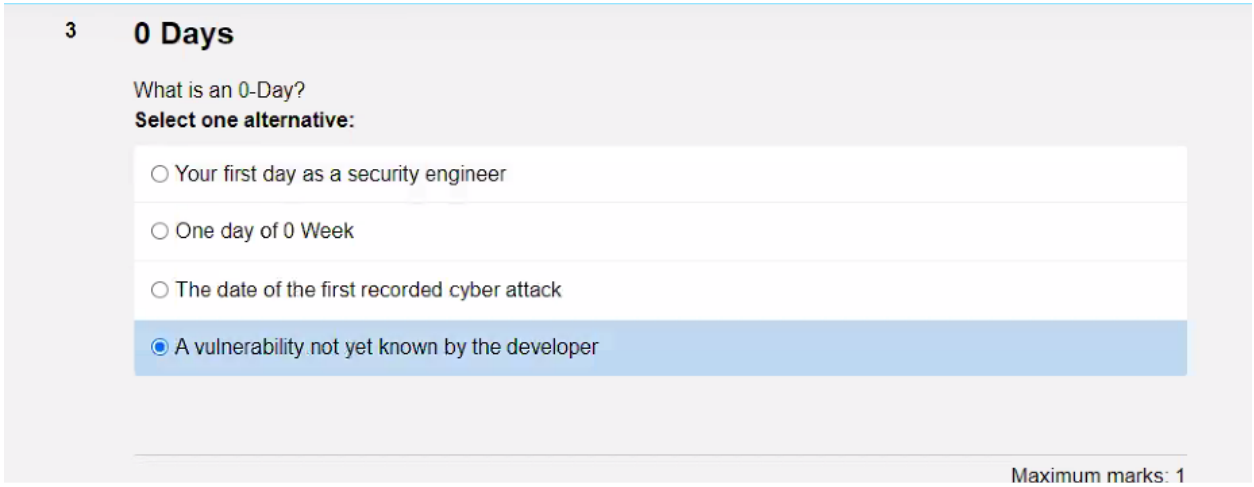
Quiz Solutions?

CAN People please collate and put some questions and answers??

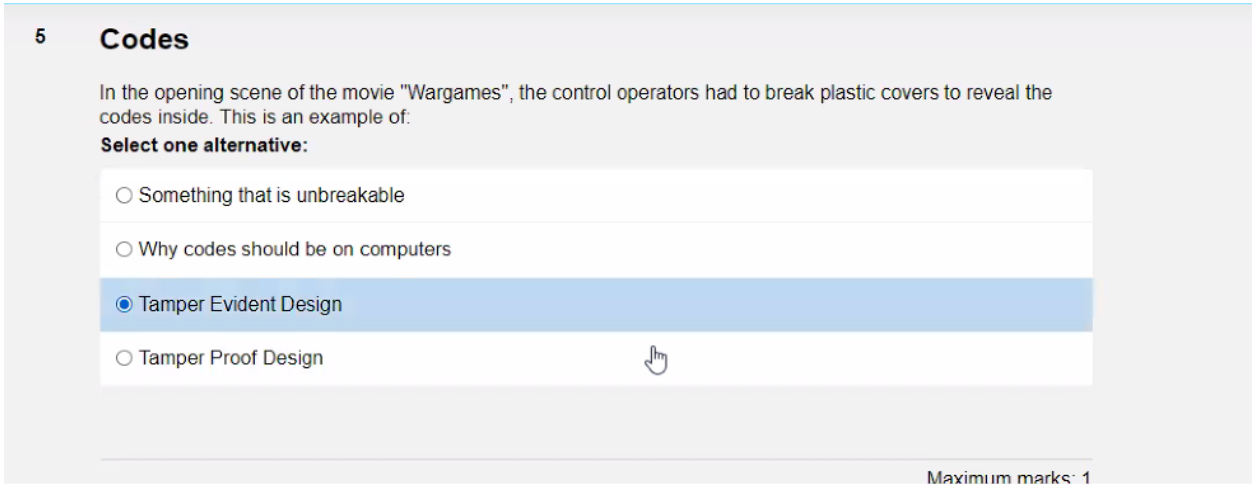
Note: I'm pretty sure most are correct, but don't take the reasoning as absolute, I may have interpreted questions right and fluked the answer. I'll try add the other quiz questions and solutions later tomorrow (1st of may at night) - Love you whoever you are <3

Week 1





Q4: most common hacker attack is social engineering = humans are the weakest point



Week 2 Quiz:

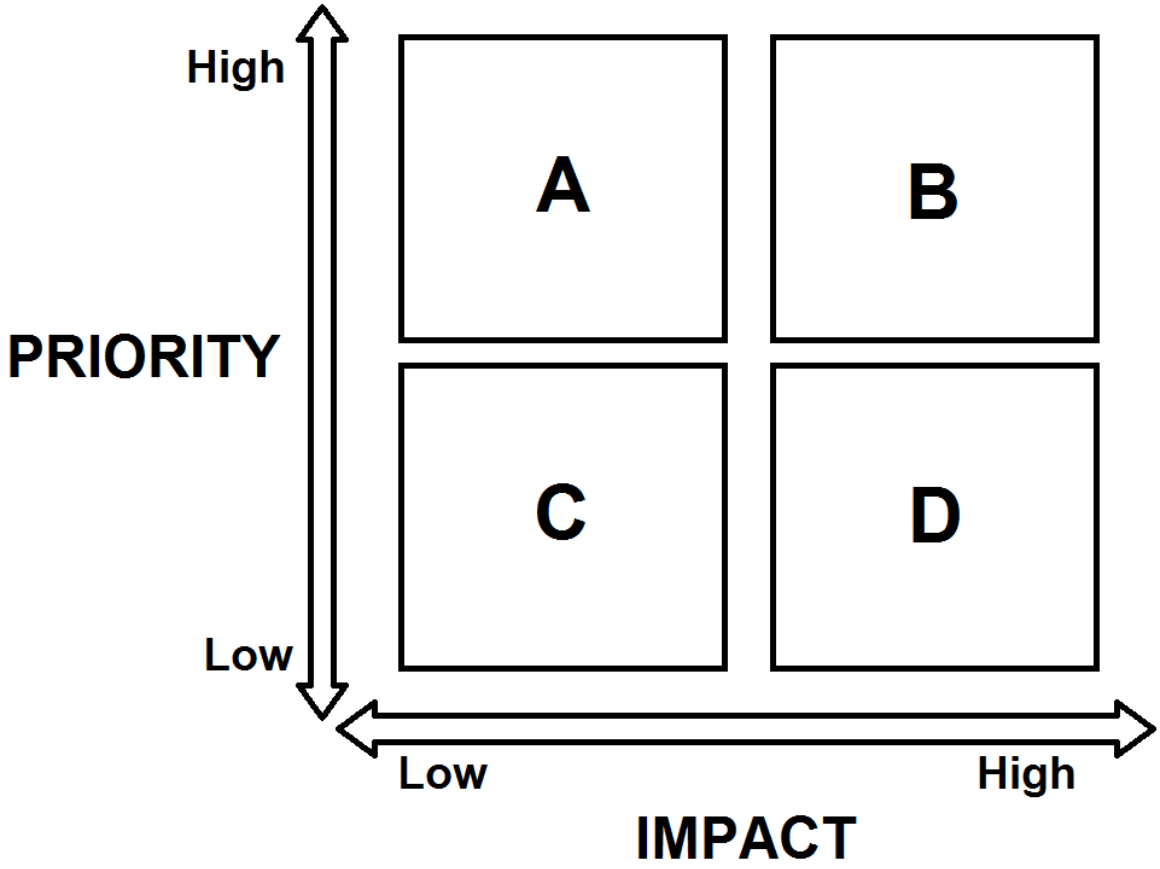
Question 1: Which is NOT an example of security by obscurity?

- Making it illegal to publish the source code for national voting software
- Encrypting a message using a standard encryption algorithm and then reversing the cipher text so the corresponding decryption algorithm won't work
- Having a security guard by the access card readers in the foyer of a building?
- Hiding a house key under a rock in the garden

TYpe IQuestion 2: You are assessing an email spam filter. If the type I error you are measuring is the proportion of emails that are spam and get through the filter, what is the corresponding type II error?

- The overall proportion of emails with spam in them
- The proportion of emails that don't have spam in them and are stopped by the filter
- The overall proportion of emails with malware in them
- The proportion of emails that have malware in them and get through the filter

Question 3: Which quadrant contains the highest priority and most important risks to deal with?



- A
- B
- C
- D
- All Quadrants

Question 4: Which quadrants contains the risks that organisations will most likely be worst at making appropriate for?

- A
- B
- C
- D
- All Quadrants

Question 5: ouQeettsiaehwsn!re

- Quote a: To know your Enemy , you must become your Enemy
- Quote b: A leader leads by example, not by force
- Quote c: You have to believe in yourself
- Quote d: All warfare is based on deception
- Quote e: The greatest victory is that which requires no battle

Solution: Quote e is the answer (deciphered)

Anyone know how to actually decipher this??

Method 1

You take the letters in chunks of three, then put them on new lines like this:

```
ouQ
eet
tsi
aeh
wsn
!re
```

And read right to left, top to bottom, and it spells out ‘Quote E is the answer!’ so you obviously choose option e. The giveaway is the capital Q and the exclamation mark. Since capitals are usually first letters and exclamation marks go at the end of a sentence, you can kind of see how it might be unjumbled.

But how did you know to do chunks of 3 letters? I basically saw ‘ouQeet’ in the cipher text and ‘Quote’ in the solution so I figured out the rule from there.

Method 2

The capital Q and ! gives away the start and beginning

This means that it is likely a transposition cipher

To solve transposition ciphers, let's put the message as a vertical column

Since Q and ! must belong in the first and last group, the minimum block length is 3

The maximum block length is also 3, because if you go 4, then the ! will end up in the second last block

o
u
Q
e
e
t
t
s
i
a
e
h
w
s
n
!
r
e

Try for 3

Put the blocks of 3 next to each other to form rows

o	e	t	a	w	!

u	e	s	e	s	r
Q	t	i	h	n	e

Rearrange rows so that it forms legitimate message (or when Q is at beginning and ! is at the end)

This turns out to be rows 3, 2, 1

Q	t	i	h	n	e
u	e	s	e	s	r
o	e	t	a	w	!

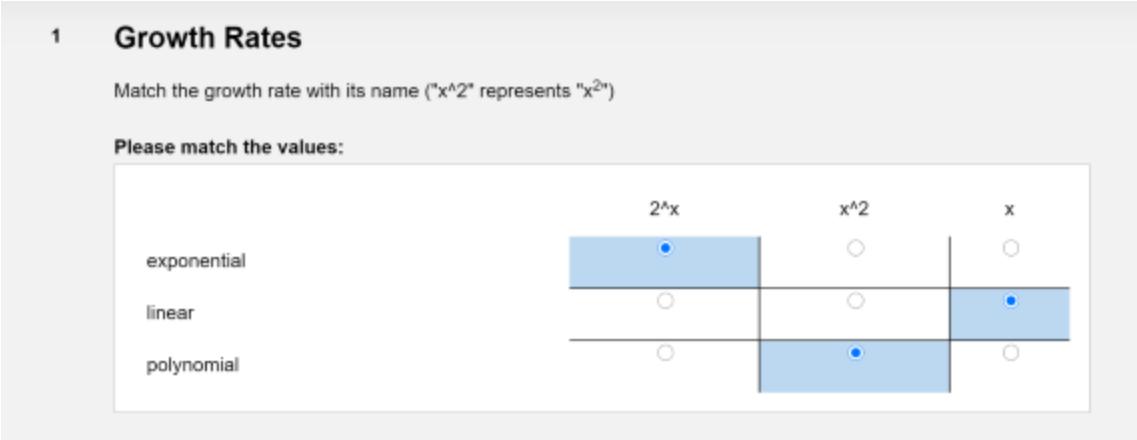
If you read it vertically:

Solution: Quote e is the answer (deciphered)

3 Week 3 Quiz:

Question 1: Match the growth rate with its name

Solution: linear is x, exponential is 2^x, polynomial is x^2



Question 2: How many binary digits does it take to represent the decimal number 1000

Solution: 2^10 is 1024 which is how many bits required and so requires 10 binary digits (can also take the upper bound of log_2(1000))

Must round the nearest power of 2 UPWARDS

Question 3: Suppose I have a password that is 6 characters long consisting of upper case letters and/or lower case letters and/or numeric digits 0-9

On average how many bits of work will an attacker need to do to brute force the password if checking a single password takes 10 bits of work?

Solution: There are 26+26+10 = 62 options.

62 ~= 64 ~= (2^6)

We have 6 combinations of these characters so we take the power of 6.

(2^6)^6 ~= 2^36. This is 36 bits of work so far, next we check the password which is 10 bits. 36 + 10 = 46 bits. However it is on average and so divide by 2

(2^36 * 2^10) / 2 = 2^46 / 2 = 2^45 and so requires 45 bits of work

Method 2: Similar approach but slightly different:

62 options for 6 characters => 62^6 many passwords available

Each password requires 10 bits of work and therefore, 62^6 * 2^10

To get bits of work, log2(62^6 * 2^10) = 45.72 ~= 46.

To get an average, divide it by half so **45 bits of work**.

Question 4:

Options: [a threat, a top man, a bug, a risk, a vulnerability, an exploit]

Solution: I made a programming mistake so the code I wrote has a bug. If an adversary could make use of my mistake to do bad things it is also known as a vulnerability

Code written by an adversary to attack me by taking advantage of my mistake is called an exploit. The adversary is to my business a threat

Question 5: Consider the following four user restrictions sometimes used to help ensure password security:

- user selected passwords should be compared against password breach databases and match
- Passwords should expire at regular intervals
- Passwords should have hints
- Complexity requirements should be used, e.g. requiring special characters, numbers, uppercase, etc.

According to the current best practice: how many of these are effective in helping ensure password security

Solution: One of them --> compare against password breach databases. Passwords should never have hints and the other two options compete with the idea of accessibility, usability and ability to remember the password and so not best practice. Security isn't as good at the expense of usability

Week 4 Quiz:

Question 1: AES keys, If 10 people all want to be able to communicate with each other privately (in pairs, ie one on one) using AES, how many AES keys do they need to accomplish this?

Select one alternative: [100,10,20,1,5,45]

Solution: AES is a symmetric algorithm and so needs n(n-1)/2 key pairs => 10*9/2 = 45 key pairs
(90 total exchanges, half of which have been double counted)

Question 2: RSA key count, If 10 people all want to be able to communicate with each other privately (in pairs, ie one on one) using RSA, how many RSA keypairs do they need to accomplish this?

Select one alternative: [100,10,20,1,5,45]

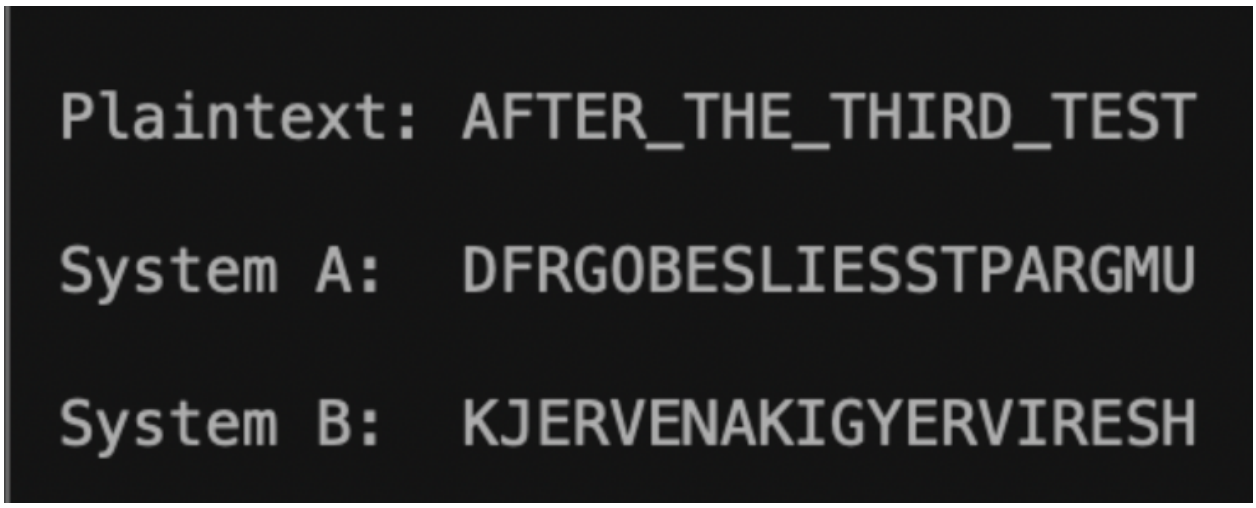
Solution: RSA is asymmetric and so only needs n key pairs, which is 10 key pairs

Question 3: Alice is sending a confidential message to Bob. What is the usual way for this to happen using RSA?

- Alice encrypts the message using Bob's public key
- Alice encrypts the message using her private key
- Alice encrypts the message using her public key
- Alice encrypts the message using Bob's private key

Solution: Public-key cryptography involves two related keys for each recipient involved - a private key (secret known only by the recipient) and a related public key (known by all senders). The sender encrypts the message using the recipient's public key. That message can only be decrypted by a recipient with a private key matching the public key

Question 4: Two different messaging systems A and B each encrypt the plaintext message below using a block cipher, with a block size of two characters per block. Also shown are the cipher texts produced by each of the two systems.



- Both A and B used ECB mode
- Both A and B might have used ECB mode
- Neither A or B used ECB mode
- A might have used ECB, B didn't use ECB
- B might have used ECB, A didn't use ECB
- A might have used ECB, B did use ECB
- B might have used ECB, A did use ECB
- A did use ECB, B didn't use ECB
- B did ECB, A didn't use ECB

Solution: **ECB, an operation for a block cipher here each possible block of plaintext has a defined corresponding ciphertext value**.

Plain-text: AF TE R_ TH E_ TH IR D_ TE ST

System A: DF RG OB ES LI ES ST PA RG MU

System B: KJ ER VE NA KI GY ER VI RE SH

The blocks will have the same value, in B we see ER and RE which are different values, as well as NA and GY when they should be the same and so know B isn't an ECB. A may be an ECB, both align and evaluate to the same blocks, however doesn't guarantee it's ECB, there are many different encryption algorithms and one could potentially produce the same result

Question 5: Suppose on average it takes: 1 second to encrypt a Merkle puzzle, 5 seconds to decrypt a Merkle puzzle if you know the key, and 10 minutes to brute force a Merkle puzzle if you don't know the key.

Alice sends Bob 10^4 puzzles so that Bob can select the key they are going to use for an upcoming encrypted chat.

Approximately how many seconds, on average, will it take Bob to let Alice know the key they are going to use? (ie measure the elapsed duration from when Bob receives the collection of puzzles to when he first replies to Alice)

Solution: On average = 601 seconds, 10*60 (second) + 1 (encryption)

I would argue that 605 is more accurate (10*60 seconds to brute force a Merkle puzzle and 5 seconds for Bob to decrypt the one Alice chose)

I would say 606 seems more accurate : 600s (brute forcing) + 1s(Bob encrypts again and sends it to Alice) +5s(Alice decrypting with thet key) =606s

Does anyone agree?

My tutor has mentioned that for this particular question, any answer between 600 and 660 was correct for this question since it was okay to take the sending/encryption into consideration when doing your calculation.

Week 5 Quiz:

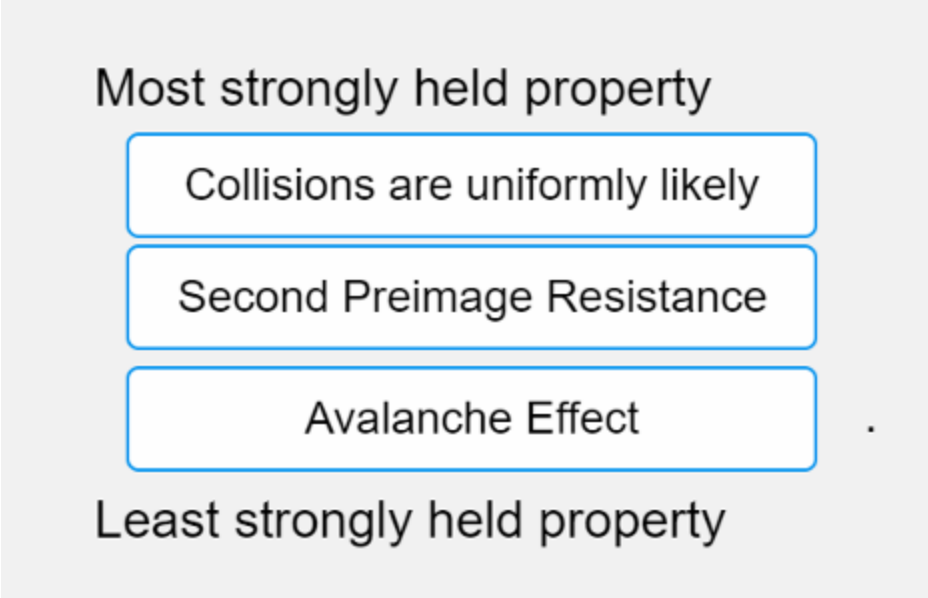
Question 1:

Consider the hashing function *h* where the hash of an input value *m* is calculated as:

h(m)=(m . mod127)

This function has some of the properties we would want from a cryptographic hash function, has some properties imperfectly, and lacks others entirely. Arrange the properties below into the order of most strongly held to least strongly held. Options: [Collisions are uniformly likely, Avalanche effect, Second pre-image resistance]

Solution: **Order:** Collisions > Second preimage > Avalanche

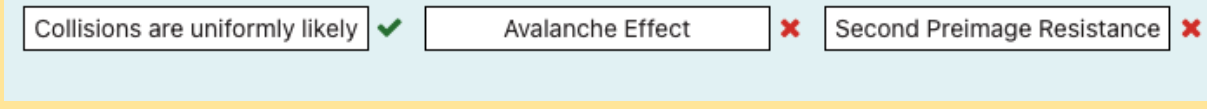


- **Collision resistance**: It should be difficult to find two different messages m1 and m2 such that hash(m1) = hash(m2). Such a pair is called a cryptographic hash collision. Collision resistance implies second pre-image resistance but does not imply pre-image resistance. Collisions are basically guaranteed as there are is only a finite number of possibilities
- **Second-preimage resistance**: it is computationally infeasible to find any second input which has the same output as that of a specified input; i.e., given x, it is

difficult to find a second preimage $x' \neq x$ such that $h(x) = h(x')$. This is unlikely to be held, since it is modulo 127 there are only 127 possible hash outputs [0, 126]

- **Avalanche property**: all modulo values to 127 lie between the field 0 and 126 and so the bits won't change, for example 4 mod 127 is just 4 and that is 00100 which is the same as the input. For large inputs for example 1101101100 which is equivalent to 876. 876 mod 127 which is equal to 114. 114 to binary is 0001110010. Comparing the bits we see that 6 of the 10 bits have flipped meaning that for large numbers the avalanche effect can exist. **11**011**0110**0 ==> **00**011**1001**0. Basically bits can flip half the time (avalanche effect is half the bits flip) in specific numbers however some numbers won't flip i.e. low numbers or specific large numbers. So this property is in the middle

- The answer to this question seems wrong? Second-preimage resistance is not held at all since there are multiple instances where two inputs will give u the same output (eg. 1 mod 127 = 128 mod 127)
    - so , shouldn't the right answer be collison > Avalanche > Second?
- Agree with this ^, when In my tute when we marked it that was the answer. Someone changed my original answer to that
- Agree as well but this answer appears to be incorrect according to moodle.
- Weird, when i did the quiz I got the question right with that order, not sure
-



Question 2: Of the below which is the strongest set of laws/rules protecting online privacy?

- The European Union General Data Protection Regulation (GDPR)
- Facebook Data Policy
- NSW Privacy and Personal Information Protection Act 1998
- Commonwealth Privacy Act (1988) and the associated Australian Privacy Principles

Solution: GDPR, it was the only one mentioned in the lecture slides, so just picked it.

Question 3: When is a hash regarded as being cryptographically broken?

- When a pre-image attack can be carried out with less work than a brute-force attack
- When the key length is too short for current computing power
- When a birthday attack can be carried in more than sqrt(n) steps
- When a preimage attack can be carried out with less work than a birthday attack

Solution: kinda common sense

Question 4: When transmitting bank funds over telegraph wires in the days of the American Wild West, which two cryptographic properties were most essential?

- Authentication
- Confidentiality
- Integrity
- Availability

Solution: ~~Can't be authentication, not part of CIA triad. Pretty sure this was right (SOMEONE CORRECT IF WRONG, IF RIGHT DELETED THIS BRACKET) WOOPS I MEANT AVAILABILITY, MY BAD~~

Authentication was part of CIA, not sure why its not the most essential, maybe authentication wasn't as important as confidentiality or integrity?

I put confidentiality and integrity for my answer and got it wrong

Integrity + Authentication was correct answer on Inspera (Buckland likes to change things afaik).

I would guess that in the exam, we would justify our choice, so if we have a good enough explanation why we chose the options we did then it should be okay.

Question 5:

Suppose there is a cryptographic hash function "UNSW60" which produces a 60 bit hash, and suppose that on average it takes 20 bits of work to hash a short PDF document (say about one page long).

How many bits of hashing work on average would it take an efficient hacker to find / construct two short PDF documents, one beginning with the word "banana", and the other beginning with the word "fish", where both documents have the same UNSW60 hash? Pick the closest answer below.

(Do not count time needed to create the documents and so forth, just count the time spent running the hash function)

Options: [50 bits, 120, 60, 30 70, 90, 80, 40]

Solution: On average, need to search half of the space. So 30 bits to find the hash, then 20 bits to make the new hash. 30+20 = 50

Wait, if it is average, and we need to divide by 2, wouldnt it be 59 bits, since 2^(60)/2?

It's not divide by 2. It's 30 because log2 (sqrt(2^60)) Nani? Thanks for the answer.

It is still a bit confusing tho. Because. It produces a 60 bit hash.. Like doesn't that mean the hash itself is 60 bit long?? But why do we care? Shouldn't it be something like it takes 60 bit of work to look through all the pdfs or smth???

I think it is like this, it took 20 bit to create one of the pdf, then using the possibility of birthday attack, it would took another 30 bit, so in total 50 bit?

But the question said constructing TWO pdf documents, shouldn't be 40?

I think in the 30 bit, while we are guessing we are also creating the pdf?

HMMMMM yeh idk :(this is so difficult

Question is bad :(oh yeh 100%. So badly worded

But 60 bit hash. So the hash itself is 60 bits in length? Thats a long ass hash?? Or do u think they mean something else ¯_(ツ)_/¯ - feels :(

Read questions carefully as they sometimes pull stuff like this.

Using birthday attack approach, it takes about 60/2 = 30 bits of work to produce one hash. The attacker wants to create two separate PDFs and hashing one short PDF takes 20 bits of work so shouldn't the correct answer be 30+20+20 = 70 bits of work?

Birthday attack - sqrt(n) bits of work - so it takes sqrt(2^60) bits of work which is 30 bits, then for each of these it takes 20 bits ea to hash so 2^30 * 2^20 = 2^50 - 50 bits of work

== ADVICE FROM THE GROUP CHAT ==

Firstly, find the amount of hash combinations

60 bit hash = 2^60 combinations

Secondly, find the average amount to brute force to match hash

BIRTHDAY ATTACKS HALVE THE SERACH SPACE

Average = sqrt(2^60) = 2^30 (because we are changing BOTH docs)

NOTE: IF WE ARE KEEPING ONE DOCUMENT THE SAME AND CHANGING THE OTHER

Average = (2^60)/2 =2^59 (because we can find our answer in half the search space for ONE doc)

Thirdly, find the bits of work to brute this average amount

Remember 20 bits of work = 2^20

2^30 * 2^20 = 2^50

Therefore, 50 bits of work on average

1 **Cryptographic Hash Properties**

Consider the hashing function *h* where the hash of an input value *m* is calculated as:
$h(m) = (m \mod 127)$

This function has some of the properties we would want from a cryptographic hash function, has some properties imperfectly, and lacks others entirely.

Arrange the properties below into the order of most strongly held to least strongly held.

Help

Most strongly held property

- Collisions are uniformly likely
- Second Preimage Resistance
- Avalanche Effect

Least strongly held property

5 **Bits of hashing work**

Suppose there is a cryptographic hash function "UNSW60" which produces a 60 bit hash, and suppose that on average it takes 20 bits of work to hash a short PDF document (say about one page long).

How many bits of hashing work on average would it take an efficient hacker to find / construct two short PDF documents, one beginning with the word "banana", and the other beginning with the word "fish", where both documents have the same UNSW60 hash? Pick the closest answer below.

(Do not count time needed to create the documents and so forth, just count the time spent running the hash function)

Select one alternative:

80 bits	50 bits	90 bits	40 bits	70 bits	30 bits	120 bits	60 bits
○	○	○	○	●	○	○	○

Consider the hashing function *hm* where the hash of an input value is calculated as:
$h(m) = (m \mod 127)$

This function has some of the properties we would want from a cryptographic hash function, has some properties imperfectly, and lacks others entirely.

Arrange the properties below into the order of most strongly held to least strongly held.

Collisions are uniformly likely ✔ Second Preimage Resistance ✔ Avalanche Effect ✔

Of the below which is the strongest set of laws/rules protecting online privacy?

- ◉ The European Union General Data Protection Regulation (GDPR)
- ○ Commonwealth Privacy Act (1988) and the associated Australian Privacy Principles
- ○ Facebook Data Policy (revision: 11 January 2021)
- ○ NSW Privacy and Personal Information Protection Act 1998

When is a hash regarded as being cryptographically broken?

- (●) When a preimage attack can be carried out with less work than a brute-force attack
- ( ) When a preimage attack can be carried out with less work than a birthday attack
- ( ) When the key length is too short for current computing power
- ( ) When a birthday attack can be carried in more than sqrt(n) steps

When transmitting bank funds over telegraph wires in the days of the American Wild West, which two cryptographic properties were most essential?

- [ ] Confidentiality
- [x] Integrity
- [x] Authentication
- [ ] Availability

Week 7 Quiz:

Select whether each of the below is a better example of authorisation or authentication:

Entering a password to login to your computer

Authentication ✔

Your boss signing a form to let you create an account on the corporate network

Authorisation ✔

Using a two-for-one code at an online shopping checkout

Authorisation ✔

Using your student card to borrow a book from the library

Authentication ✔

Which legislation in Australia best allows citizens to request access to data held by Federal Government Entities?

- (●) Freedom of Information Act (1982)
- ( ) Government Information Public Access Act (2009)
- ( ) Data Availability and Transparency Bill (2020)
- ( ) Assistance and Access Act (218)

A multi-purpose IT company called MEGA has decided they will offer a service to the world which will sign arbitrary documents to assure integrity.

MEGA sign documents by encrypting them with their private key, so that anyone may then decrypt them using MEGA's public key (which is published to the world).

MEGA make money by offering a service where anyone can pay to have data backed up without requiring access to the internet. The process for this is the customer encrypts the data they wish to back up with MEGA's public key, stores it on a hard drive and sends the hard drive to MEGA by courier.

You discovered the company uses the same key pair for signing as it does encrypting.

Which keys would an attacker need to brute-force to successfully acquire a company's data, assuming they were able to intercept the encrypted data en-route to MEGA?

- A new set of keys
- The company's public key
- The company's private key
- The company's key-pair
- None ✔

Why is the answer none? Is it something to do with the fact that they will sign an arbitrary document?
Because the company's public key is published to the world, the attacker doesn't need to brute-force any key.

Okay so if you wanted to use MEGA's integrity service, you'd have to use MEGA's public key, encrypt, send to MEGA

Then MEGA uses private key to decrypt

But the major flaw is that MEGA also has a separate service that verifies documents by encrypting them using their private key, to keep the documents safe

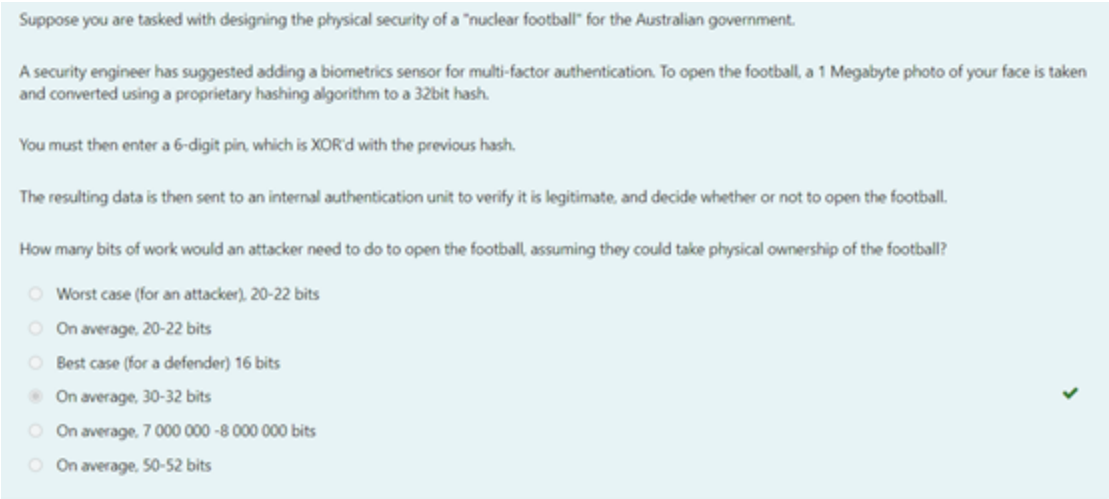
If attacker intercepted a client's package, it is just data encrypted with MEGA's public key, which means the only thing that can decrypt it is MEGA's private key

Attacker could ask MEGA to sign it with their private key (posing as someone who wants them to sign it to keep it safe) and that would decrypts what has been done

This basically decrypts the package because they use the same key pair
No effort needed

anyone know why committee is a single point of failure? - they're meant to be benefits so I;m taking it as a benefit is that a committee is not a single point of failure? - classic
What does arms length oversight even mean???? It means overseeing a group of people from a distance and without participating

^^ agreed this question is so poorly worded richard pls dont do this in exam



-

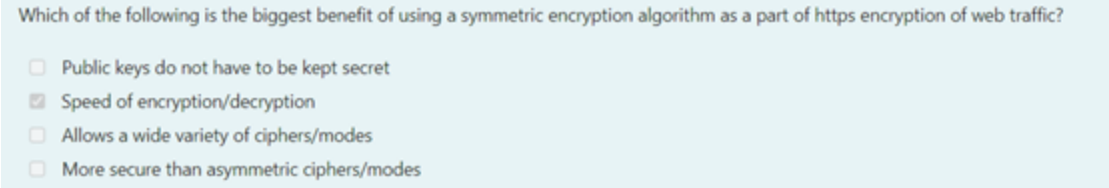
The explanation for this is (I believe) if an attacker is to brute force the code, since the 6-digit pin is XOR'd with the 32 bit hash, the result is 32 bits, to brute force 32 bits on average will take 31 bits of work.

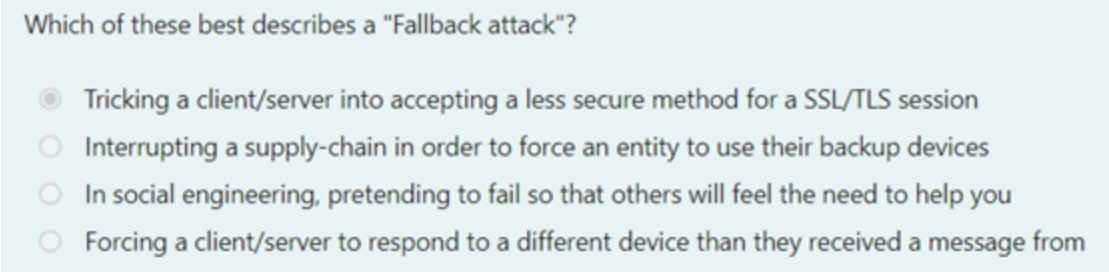
- **Yeah I had the same idea. The 6 bit code is irrelevant, since XORing it with 32 bits still produces a 32 bit hash, so overall there are 32 bits needed to be worked out, and so on average it would be 32-1=31 bits of work to do.**

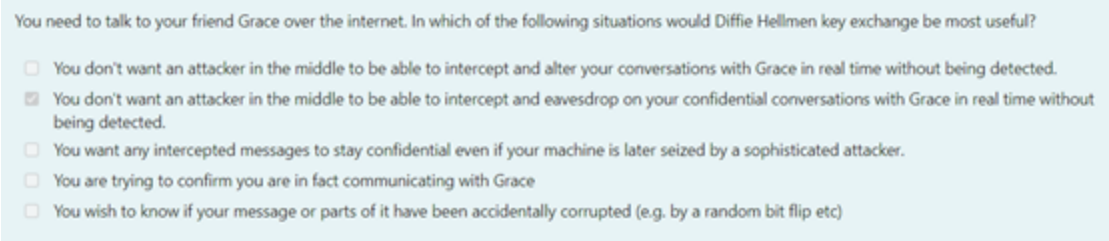
-

-

Week 8 Quiz:





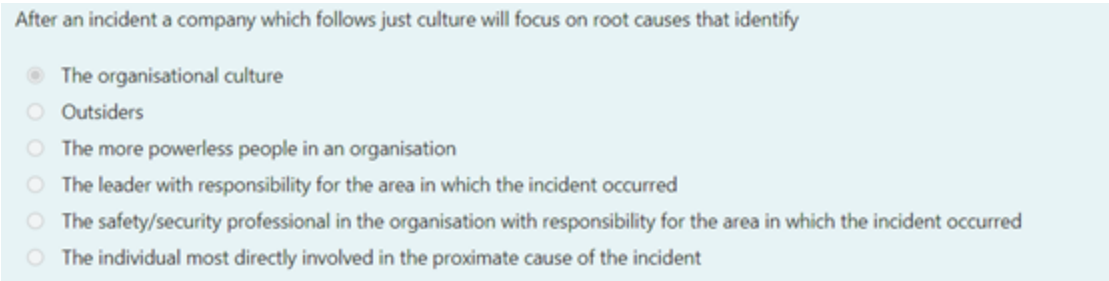


There are 3 ways of key exchange we've learned:

1. Merkel(not modern)
2. Encrypting AES key using public key and the other side can decrypt with private key hence established a shared AES key
3. DH key exchange

But in my opinion, the advantage of 3 over 2 is forward secrecy, you can use method 2 to avoid eavesdrop too. Why is the answer not the 3rd choice??

I agree with this ^, DH is known for forward secrecy. DH can be intercepted (man in the middle)

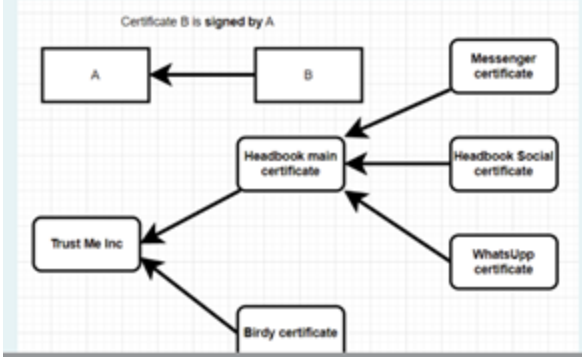


Trust Me Inc is Certificate Authority who signs certificates for two social media companies *Headbook* and *Birdy*.

Headbook uses their private key associated with the certificate signed by *Trust Me* to sign certificates for three individual apps / websites they own: *Messenger*, *Headbook Social*, and *WhatsUpp* (as shown in the diagram below).

Birdy only has one service, which uses the certificate signed by *Trust Me Inc*.

Service	Number of users
Messenger	1 million
Headbook Social	3 million
WhatsUpp	0.5 million
Birdy	2 million



How many users are affected if the private key associated with WhatsUpp's certificate is leaked? 0.5 ✔ million.

How many of the above users are affected if Trust Me Inc has their private key leaked? 6.5 ✔ million.

How many users are affected if the private key associated with Birdy's certificate is leaked? 2 ✔ million.

How many users are affected if the private key associated with Headbook's main certificate is leaked? 4.5 ✔ million.

2020 Exam Solutions

=== Question 2 ===

How useful: [1 number between 0 and 10, 0 meaning not at all useful and 10 being incredibly useful]
8
Strengths: [50-250 words]
It provides some simple steps that people can carry out to stay safe online. It also contains a link to additional resources such as the government stay smart online site, and the uni IT department. It also contains a clear headline outlining the goal of the email message which is 'staying cyber safe at university'.
Weaknesses: [50-250 words]
It contains a lot of text, and people are likely to skim read or ignore it altogether.
It tells people to not click on suspicious links, then immediately provides a link to click on for 'free Antivirus Software.
Puts blame solely on the students and staff rather than taking any responsibility themselves.

Your alternative email: [100-500 words]
(bullet points are fine no need to write the whole thing just let us know what it contains)

- - start off by explaining what happened at Melbourne Uni
- Clearly tell people what data was lost/could have been lost if you know.
- - reassure the Random Uni staff/students that we have the necessary controls in place, but need everyone to do their bit to help
- - provide a checklist for everyone to see and tick off the things that they have done already or might still need to do to stay cyber safe.
- - provide a short list of don'ts with cyber safety (e.g. don't click links in emails unless you trust the sender)
- - provide contact info for the Random Uni IT hub, and links to resources such as the staysmartonline gov site.

Explain the main differences about your email: [50-250 words]
The main difference with my email is that it contains a **checklist**, which is visually much easier to interpret than a few paragraphs of text telling you what you should do. The checklist would contain things like 'I have updated my computer to the latest edition', and 'I use an antivirus program on my computer', or 'I don't use any software from untrusted sources' etc, etc. This way people can be engaged with the message being made in the email, and by making a personal connection with the first person checklist, people are less likely to ignore it.

=== Question 3 ===

Q3.1 Should there be a separate Cyber Operations Center or should there be one combined Operations Center for all incident and emergency response? [1 word SEPARATE or COMBINED]
COMBINED
Justify your decision [15-75 words]
By having the cyber security staff in the same room as the other staff they can quickly and easily work together to form a solution to a problem. Another benefit is that the cyber security staff are not isolated from the rest of the team, and are able to stay up to date with any information as it comes in to the center. By having a combined operations centre, security staff will get a seat at the table all the time, assuming that each department in the operations center has at least 1 representative.
List and explain any insight(s) from Apollo 13 Mission which contributed to your decision above [25-125 words]
In the film, whenever there was a decision being made in mission control, they had several people put in their opinions from their respective disciplines (e.g. medical, lunar module, propulsion etc), which allows for a broad range of voices and would not be possible in a separated working environment.

Q3.2 What are the 3 main insights that the Apollo13 Mission provides which are relevant to how you will DESIGN the center? Most significant insights first. Don't repeat insights already given in Q3.1 above.

1.
Insight [10-75 words]
In mission control there was the large control room for the majority of team leaders/management to operate, but also smaller specialised areas for each team to meet/prototype etc.
How it affects the design [20-100 words]
By designing the control room to have at least 1 representative from each division this ensures there is a voice from each team being heard in discussions. If needed, these leaders can meet up with their fellow members in a smaller space and discuss and design things. This ensures that teams have the freedom to hold their own meetings for their own personnel and have access to a specialised area for their team (e.g. cyber sec team would have computers pre configured for their line of work, flight engineers might have a workshop with catalog of parts for aircraft and tools to test parts with).

Q3.3 What are the 3 main insights that the Apollo13 Mission provides which are relevant to how you will OPERATE the center? Most significant insights first. Don't repeat insights already given in Q3.1 above.

1.
Insight [10-75 words]
NOT ANSWERED
How it affects the operation [20-100 words]
NOT ANSWERED

===Question 4===

To what extent? [one letter, A or B or C or D - see above for meaning]
B

Justify your answer: [60-300 words]
Depending on the encryption method used, it will significantly improve the security nature of the emails. Yahoo was able to detect the breaches,
proving that it was tamper evident, as their security is not high, I would encrypt it to the maximum possible standard to 256 bit, which significantly reduces the risk.

- 256 BIT encryption
- Includes hashing
- Tamper evident (Another one), check from the receiver end. If the file was modified by middle man

The issue with encryption is that you need a shared secret that both people know in order for it to work. As you're sending an email to somebody, they are probably somewhat unfamiliar to you or you know them on professional terms rather than personal. Unless you pre-establish some sort of method of having keys, such as a shared list of nonces that you both have and use in communication, you will need to also send the key using some method through the email.
You could use merkle or Diffie Hellman, but both these methods will take a long time and be inefficient to do through email unless you were able to somehow automate it. You could do asymmetric encryption though, which would possibly resolve this.

===Question 5===

What is the name of the UNSW CISO? [0-2 words] (*Assume you are researching this in 2020*)
Peter Cooper

Name one significant (but not personal or intrusive) thing about the UNSW CISO you learned in your recon? [0-75 words]
. Reach Goal by passive recon, by not engaging the desired target.
. collecting info from the already present given in the internet
. Previously worked at GWA group and made significant improve on the company (LinkedIN)
Clearly explain a simple and safe social engineering strategy an attacker could follow to have the CISO visit a webpage they control. (To be clear - don't actually do this in the real world - you could face charges and/or be expelled from uni). Assume the attacker has roughly the same access and resources that you yourself have available to you. [80-400 words]
. Gain Information about the intended target from the previous place he worked at.*

- Send a phishing email saying the UNSW secsoc wants him to present at their event
- Disguise his email as the dean? Potentially
- Email/advertisement for rock climbing gear
- He seems interested in conference presentations, disguise as a conference manager requesting him to present
- Pose as a member of UNSW IT who requires some of their personal info to do XYZ
- ADD MORE

Gain information from outside sources to see if it can be used for

- Potentially same password is used for multiple passwords
- Try to infiltrate the weaker sites

=== Question 6 ===

Q6.1

Potentially - when an email app opens a message with an embedded image, a lot of information is sent to the server that's hosting the image

This information can include an IP address, device type, operating system version, geographical location, screen size, device language, device time, and much more.

Do people agree with this?

Possibly XSS as well? (COMP6841 content)

What I had written: “Unlikely. While it is possible for attacks to be conducted via images (e.g. XSS scripting), this can be prevented if the email client (or web-page) sanitises input correctly and properly controls script usage. If this attack did occur, it might compromise the email account (e.g. attacker grabs session token, changes password), but it would not impact the device unless there was also a sophisticated (probably 0-day) attack against the browser/email client. So unless there’s some serious vulnerability in the email client they should be safe.”

Just IP address and user agent info is pretty trivial stuff, it gets exposed to every website you visit and more besides. I wouldn’t say that an attacker getting that info make the friend ‘at risk’

Q6.2

Should we use Method B next year? [yes/no]

No

yes

Why? [60-300 words]
Multiple layers of encryption does not usually increase the level of security, but rather, slows down attackers. Encrypting a file with password A and then encrypting again with password B offers the same security as encrypting with a password that is a combination of A and B.

2 layers is unnecessary and wouldn't measurably improve security vs 1 layer done properly. It would add complication in encryption / decryption, which would: increase chance the process wasn't done correctly, thus potentially introducing vulnerabilities; and more likely, cause issues for students when it came time to do the exam.
Step 2 alone should be sufficient (AES 256 is good, although ideally CBC would be avoided).

Agree with this - better to just use a strong random key than nested encryption

Q6.3

(Answer written in white to not spoil)

I had a different answer:
I rate tthe first answer cause it says there is only one rabbit with no twin i dont think a rabbit with a hat makes it a twin. *Hmm fair point, I guess they don't consider accessories to be part of it? Perhaps its more a question about describing how you would solve it for a larger scenario rather than this one (i.e. all marks for analysis, no marks for answer)*
Here's my attempt at a scalable method
Split each bunny into sections, and assign each section a bit, 0 if it isn't shaded, 1 if it is.
[left ear][right ear][left eye][right eye][mouth][throat/bit between mouth and tummy][tummy]
Then translate bunnies to binary numbers

1000000	0000101	1101000	0000100	0000000	1101000
1111000	0000001	0000111	1000000	0010000	
0000000	0010000	1100000	0011000	0110000	0001000
0110000	0011000	0000101	1111000	1100100	0100000
0001000	0100000	0000111	0000100	1100000	0000001

Translate binary numbers to dec

64	5	104	4	0	104
120	1	7	64	16	
0	16	96	24	48	8
48	24	5	120	100	32
8	32	7	4	96	1

Then sort the number and find which occurs once. (The translation to dec isn't necessary but makes it easy to find the odd one out)

> ^ You prob dont even need to sort it - I would just iterate through the list of bunnies and would add them to a set.If it is not in the set or if it is already in the set, remove it. This means twins will not exist in the set after going through all the bunnies leaving the twinless one in the set.

My method involved sorting out rabbits by certain patterns (similar to your binary system) and marking them off as I see it.

Here is my method: Pick one feature (for example: brown right ear) and count how many rabbits have that feature, if the number is odd, the non-twin rabbit must have that feature. Repeat this process for all possible features, and eventually, you will get a unique description (set of features) of the non-twin rabbit.

=== Question 7 ===

Is it possible for the App to work as promised if the database was just 10 Megabytes in size? [One word - YES or NO]
NO

Justify your answer, include calculations [25-125 words]

25 million people (population of Australia) = (2^5)*(2^10)*(2^10) which is approximately equal to 2^25. (You can just use wolfram alpha to convert 25 million to binary and the number of digits is the exponent). Since we need to uniquely distinguish between 2^25 possible values, this means we need each person's hash to be 25bits long.
25 bits is approx 4 bytes (allows some overhead).
4 bytes * 25 million people = 100MB.
Even if we assumed 3 Bytes per person this would produce 75MB, still more than the 10MB stated by them.

- I think I'd disagree with the above answer because the length of the hash does not need to be that long. The hash is a small fixed length.
    - I don't think the hash necessarily has to be 25 bits because that assumes that every person in Australia has a license when it's probably around 50 or 60% (not sure). BUT I think using 25 bits would future proof it somewhat. The reason it needs to have so many is that if you use not as many bits, there are going to be guaranteed collisions as the output range of the hash function is less than the input domain so with say 25 million inputs and only 5 million possible outputs, there will be a significant number of collisions. If you have a 25 bit hash, there could still be collisions but it is much rarer.
- Can you explain a bit more? Do you have any alternative working? If the hashes were any smaller then there would be collisions and the system can't map a face to a single driver's license?
- Hash collisions are inevitable, but extremely rare.

What is the switchover value? How small could X be and have it be possible that their App could work as promised? [one number]
100MB
Justify your answer, include calculations [75-750 words]
Same working as above. This assumes there needs to be a way to uniquely identify between 25 million Australians, so a hash of 25 bits length can achieve this.
25 bits takes 4 bytes to represent it.
4 Bytes per person * 25 million people = 100MB of data.

I also accounted for the 8 digit drivers license number which has 8^10 possible permutations, rounded to 4 bytes for storage and assumed that only 60% of the population has a driver's license, which left me at about 120mb.
I checked the figures on license holders and I think it's actually about 70% of the population.

=== Question 8 ===

I guess people add their own to this list/ cross out ones that aren't suitable and comment as to why they aren't:

List your recommendations about what he should do? [100 - 500 words]
- Install and run a trusted antivirus program such as malwarebytes/Norton/Kaspersky etc
- Visit a trusted friend's house to login to his accounts that have been breached and change his password there
- Enable 2 factor authentication on all his social media accounts and websites accounts he has (if not already enabled), using an authenticator app rather than SMS where possible
- change the recovery email to a new email account by setting up a new gmail, proton mail, or other provider, just for password resets
- Remind friend's dad that he should not click links or download attachments in any emails
- use passwords that meet NIST standards
- Could be an insider attack, so change the password of the PC and ensure that it is locked whenever not in use.

Explain your reasoning behind making these recommendations. [160-800 words]

1. The antivirus program scan will hopefully be able to detect and remove malware that attackers may have installed on his machine from him clicking malicious links. This will make the computer safe to use again.
2. By visiting a trusted friends to reset the password, this reduces the risk of changing it on his potentially infected computer (that could still have the antivirus scan running on it), and hackers obtaining his new passwords with something like a keylogger
3. Enabling 2 factor authentication will reduce the chance of an attacker knowing the password being able to log in and do the same again, as they would need to provide the second factor which only the dad has. The phone SMS method is not 100% secure as hackers can spoof a number and still retrieve the codes to login, so the authenticator app is more secure.
4. Changing the recovery email means that if the original email account used for recovery was compromised, then the new one will be a clean slate and is not compromised hackers would have to access this one to to hack in again
5. Reminding him to not do this reduces the chance of infection
6. By creating passwords that are long, meaningful and do not contain hints reduces the chances of hackers brute forcing or guessing them, and will remove the need for him to write the password down, which can be bad as someone could break into his house and find it.

=== Question 9 ===

Suppose internet giant Gitzon is considering having its new authentication system store password hashes produced by a custom secret hash function (so that existing public rainbow tables can't be used to crack passwords).

Do you recommend that they adopt this new approach? [1 word YES or NO]
Yes (Id say no)

Briefly justify your answer [60-300 words]

- Better than if u used public rainbow tables? - thoughts ?
- I would say No - because this new function has not been subject to testing/ time and as such, it is unlikely to be secure? Thoughts?
    - “Don’t roll your own crypto”? But this is for hashing, so does it still apply?
    - I reckon
    - Security through obscurity? They clearly aren’t willing to show their hashing algorithm to the public too
    - They should just use a salt and a publicly accepted hashing algorithm.

If they're a true internet giant they might have the resources to do this properly, but such an expenditure would be wasted effort given the existence of proven open source cryptographic hash algos already. Rainbow tables can be effectively mitigated using salt, no need to reinvent the wheel here!

Suppose Gitzon adopts this new authentication system, and suppose you are advising the Prime Minister who uses Gitzon is worried that some attacker may target him and break into his personal account. To what extent would it be harder for the attacker to find his password if Gitzon stored salted hashes instead of just storing the hash directly?

[A] It would make it much harder for them to find his password
[B] It would not make it much harder for them to find his password
To what extent? [1 letter, A or B as described above]
A

I said B: It shouldn't matter, since attackers would not be able to pre-generate rainbow tables anyway (they don't have access to the new secret hashing algo, it is secret per the question). Unless you assume that the algorithm will eventually be leaked, in which case A.

Briefly justify your answer [60-300 words]
By adding salt can protect against different attack vectors such as hash tale attacks while slowing down dictionary and brute force offline attack s

=== Question 10 ===

- Can people please help with q10 - i have no idea. Yeah same :/ feelz
- Written so preliminar thoughts- any discussion would be great :)
- I think we want to choose methods of encryption that are not too easy, but also not too hard to crack, since Merkel puzzles rely on it being relatively easy to brute force one, but it takes too long for an attacker to brute force all of them? Does anyone agree?
    - I agree :)

10.1 One Time Pad
Suitable? [1 word, YES or NO as described above]
No

Why? [30-150 words]

You won't be able to crack it unless you know the key. Ie. can't crack with brute force
A one time pad requires both the sender and receiver to have the same one-time pad otherwise they wouldn't be able to encrypt/decrypt each other's messages; also raises the question, how do they safely share the one time pad? How does the sender know that the receiver has the correct one time pad or is authorised to have that one time pad?
Anything can be cracked with brute force. An OTP is just a key that's as long as the plaintext. Assuming each message is about 80 ASCII characters, it would take 256^79 = 2^(8*79) = 87 bits of work on average to brute force crack a message. That is a lot of work though, might be infeasible for this situation..

- But I thought that there is no way to crack this unless you know the key
  - ^ this is true as long as the key is longer than the message
  - By definition a OTP has a key as long as the message and thus is unbreakable assuming they followed the rules
- Because without the key, there is no way to prove that whatever potential decryption you've arrived at is the correct one.
  - Theoretically it's possible to randomly find the OTP but this is a brute-force attack and like what above said you can't be sure if the message is correct either

10.2 Caesar Cipher
Suitable? [1 word, YES or NO as described above]
No

Why? [30-150 words]
Way too easy to crack
At the basic Caesar Cipher, there are only 25 possible combinations for the cipher (26 letters in the alphabet - itself). So it can be easily broken by computers. On top of that, because it only uses one unchanging key (how many letters you move up/down) there will be patterns that frequency analysis can pick up.

10.3 Vigenere
Suitable? [1 word, YES or NO as described above]
Yes
I'd say No
Why? [30-150 words]
Reasonable difficult to crack a single one
Mainly because the receiver has to have the key to decrypt the message and it's weak to frequency analysis etc. Since we're assuming for the Merkle puzzle it's probably appropriate.

I'd also say no. With vigenere, you can work out the key very easily if you know the plaintext and ciphertext (just modular subtract the corresponding letters), and all the messages start the same way. So once you crack one puzzle, you get the first 15 characters ("This is puzzle ") of the key for free for all the rest of the puzzles.
Have to agree too. Vignere ciphers can still be broken by a chain of methods - once you work out the length of the key, frequency analysis can be used to find the actual key. Other known methods also exist

- So no because it's too easy to crack?
    - Basically

10.4 RSA (encrypted using a 512 bit public key)
Suitable? [1 word, YES or NO as described above]
No

Why? [30-150 words]
Quite hard to crack one, let alone, many. RSA is also known to be very secure

- I would probs say No, this is not suitable since its too hard to brute force one of these for a merkel puzzle
- Yeh ok, i agree.
- No, because RSA is a very secure encryption that takes even supercomputers years to crack. We want the puzzles to be fun, not to be time-and-resource consuming.
- the keys to be

Unless Richard has a supercomputer, 512 bit RSA keys are probably too big to brute force within a window of a few hours (so he can communicate before the exam). A 200-300 bit RSA key might work.

10.5 SHA256
Suitable? [1 word, YES or NO as described above]
No.

Maybe, but probably not

Why? [30-150 words]
This is a hashing algorithm - not encryption. SHA-256 is pre-image resistant. Once it's hashed, you cannot "decrypt" it since it is a one way function it's practically impossible to reverse

If Richard knew the Merkle puzzle text format ("This is puzzle x, the key is y"), it might not be too hard to setup a preimage attack and brute-force one of the puzzles, since most of the message would already be known ("This is puzzle one, the key is ___"). The key is 12 bytes, so as far as I can

see it becomes a question of whether you can feasibly hash 2^96 of the puzzle-candidates (or on average half that). That said, this keyspace might still be too big for an effective preimage attack.

=== Question 12 ===

(Answer written in white to not spoil)

How do you solve this in like 15 mins?? Any tips?
What I looked at were words that include the starting letters and what was really constraint. For example: 1,9,3,3,5,3,15 → there are not many words that can fulfill this structure. Then you just go from there, does that kind of make sense?
Use frequency analysis, i.e. one of the numbers occurred about 28 times (the most) → most likely the letter “E”
Also can use double letters
Look at the longest words, often they will have the fewest possibilities
Also if they're certain sizes, they can only really be certain numbers (like say 1-3)

=== Question 13 ===

=== Question 14 ===

Just an example:

Name the example and roughly when it happened [10-50 words]
NSA Clipper Chip, mid 90s

Briefly describe what happened [25-125 words]
The NSA developed an encryption chip marketed primarily for use in telephones. It used a poorly-designed implementation of a proprietary public-key algorithm, and the private keys were shared with the US government to allow them to listen in as they liked. This 'feature' quickly became known and as a result no-one wanted to use the chip.

Briefly state how this demonstrates the issue [60-300 words]
It proves that you still have to trust the underlying hardware that you are using, and thus the companies and people who have designed and built them

2016 Exam Solutions

===Question 0===

===Question 1===

Assume the passport photo is 200x300 px and is black and white.
Each pixel will have two possible values black or white, i.e. a binary value.
Hence there are 200*300*1bit = 60,000bits of information in the image
60,000 = 6*10^4 is approximately 2^16, therefore 16 bits of security

~~**If you consider a bit of information to be equal to bits of security, then the answer could also be 60,000 bits of security? (Will update this answer once my tutor replies to email)~~
~~- any updates ?~~
Yep so my tutor confirmed that it should be 16 bits as the answer, based on the assumptions made. My tutor said *"Bits of information is the space you need to store something, while bits of security relates to the work an attacker would have to do to recreate the information"*. They also said you could improve the answer by using RGB instead, so 24 bits per pixel, but the above answer would be acceptable.

Could you also tackle this question from a hash perspective? (i.e. assume the thumbprint scan creates a 128 bit hash, hence the bits of security is 2^128 = 128 bits)

From a paper I found on the internet, it's a bit more complicated
This is because, because analyzing the entire fingerprint would be time and resource intensive, most digital fingerprint readers would only look for certain "points of interest". As a result, the number of people on planet Earth becomes somewhat irrelevant for this calculation. So instead, guessing that most fingerprint readers will take about 30-40 points of interest, and that if we decide to store the info as x-y coordinates that takes up 8 bits, we get 640 bits of security

=== Question 2 ===

Suppose you were part of the original HTTPS protocol design committee back in the dim distant past. Why would you not use RSA to encrypt a HTTPS session
RSA takes too long - anything else?
Does not offer perfect forward secrecy.

How could you achieve Perfect Forward Secrecy in an HTTPS session?
Using diffie-helman

RSA is simply too slow in comparison to symmetric algorithms. Better to use RSA (or Diffie-Hellman key exchange) initially to share the keys for a symmetric algorithm and then use the symmetric algo from then onwards. This is basically what TLS does.

=== Question 3 ===

1. Alice and bob does their own computations
   Alice: 2^5 mod 33 = 32
   Bob: 2^8 mod 33 = 25
2. They exchange answers
   A->B 32

B->A 25

3. Alice takes bobs number and raises it to her secret power and mods it by n
   Alice: 25^5 mod 33 = 1
   Bob: 32^8 mod 33 = 1
4. They compute the same number!

The above answer makes no mention of Mallory (the man in the middle). If M is intercepting the Diffie-Hellman key exchange (assume her private secret is 9), then shouldn't the answer be:

1. A -> M 32
2. M -> A 17
3. B -> M 25
4. M -> B 17

Yep agree

Just to clarify for people, the result of a mitm is that we get a shared secret between A and M, and a separate secret between B and M, but A and B both think there is a single session between A and B.

assumption that M knows
- A->B 32
- B->A 25
- m=33
- g=2
- and that M can block messages from reaching the other party

Where does 17 come from? => (2)^9 mod 33 = 17, Mallory's private key is 9

=== Question 4 ===

- Malware could be somewhere else
- some infections can actually write themselves to your RAM, which is unaffected by an OS install. Also, some infections can inject themselves and reflash your BIOS, which is also unaffected by an OS install
    - Writing to ram wouldn't help much since system RAM is volatile memory (all data is lost after every restart), and wiping the drive would usually involve a restart. Writing to flash on the mobo (which is non volatile) might work, although we're talking only a few mbs

- Can also install themselves into graphics card and sound cards, as those also have some data storage
- hardware has the backdoor built into it.
    - Certainly a possibility - look into the intel management engine and AMD PSP
- Could the malware be written by the manufacturer (by mistake) onto the ROM? In that case, no matter how many times you wipe your system, the ROM won't be, and will always have tht malware? Chilling thought if true!
    -

=== Question 5 ===

You chair the government panel considering the introduction of new mandatory data breach notification requirements for organisations and you have just finished reading all the recent public submissions. What are the main arguments for and against introducing such a requirement?

For:

- Allows users that are affect to take immediate and appropriate action
- Encourage organisations to increase security
- Save some face before the data leaks out
- Makes businesses and other services aware of possible followup attacks

Against:

- Reputational damage for entities involved in breach
- Increased costs for organisations to increase security measures
- Crackers would know

=== Question 6 ===

Can someone check if this is right:
Pseudorandom (check if I'm wrong)
If the length of IV is n bits, then a birthday attack to find common bits will require n/2 bits of work (since each bit is either 0 or 1). Since 2 billion messages were checked before one collision was found (around 2^31 messages), around 31 bits of work was required to find n/2 bits, so the total length of IV is 62 bits.

- I agree
- Yep agree also, solve for n in sqrt(2^n)=2*10^9 = ~62 bits

=== Question 7 ===

- Length extension attack
- SQLi
-

=== Question 8 ===

Many operating systems and antivirus systems now give preferential treatment to code which is signed (for integrity and authentication) using an X.509 digital certificate. So malware authors would like their malware to be signed. List four ways they could manage to have their malicious code signed. If you can think of more than four, list the four most important first and the other ones below.

1. Generic chosen-method – In this method C tricks A to digitally sign the messages that A does not intend to do and without the knowledge about A's public key.
2. Direct chosen-method – In this method C has the knowledge about A's public key and obtains A's signature on the messages and replaces the original message with the message C wants A to sign with having A's signature on them unchanged.
3. Known-message Attack - in the known message attack, C has few previous messages and signatures of A. Now C tries to forge the signature of A on to the documents that A does not intend to sign by using the brute force method by analyzing the previous data to recreate the signature of A. This attack is similar to known-plain text attack in encryption
4. 3. Key-only Attack - in key-only attack, the public key of A is available to every one and C makes use of this fact and try to recreate the signature of A and digitally sign the documents or messages that A does not intend to do. This would cause a great threat to authentication of the message which is non repudiated as A cannot deny signing it.

Maybe try inserting your own key into the OS trusted cert repo? A local attack would probably be mt feasible

=== Question 9 ===

Prove, showing your calculations and assumptions, whether or not 128 bits of work is too much for an attacker to be able to do, and make a recommendation as to whether snapchat should switch or not.

Assume attacker uses a computer that runs at 5Ghz and has 4 cores, so there are 5*10^9*4 operations per second
= 2*10^10 = approx 2^31

So there are approx 31 bits of work being done per second
There are 60*60*24 seconds in a day = approx 2^17
So in one day an attacker can do 31 + 17 = 48 bits of work

To complete 128 bits of work:
2^128 / 2^48
= 2^(80)
= 1.2 * 10^24 days

On average you only need to brute force half of the sample space, so half the time taken
Average time to do 128 bits = 6 * 10^23 days (This is a very very long time)
So Snapchat should be safe to use 128 bit AES as 128 bits of work is too much for attackers to do in a reasonable amount of time.

Recommendation: Snapchat stays with the 128 bit AES as it is enough to stop attackers from brute forcing the encryption. Changing to 256 bit AES is more secure, but is not necessary, and takes longer to encrypt/decrypt and uses more computing power.

What are the main two security weaknesses that this code reveals?
Weakness 1: Uses AES-128-**ECB.** ECB mode is generally discouraged.
Weakness 2: The key used to encrypt and decrypt is in the code. (not sure of this)

The fact that the exploit can decrypt all snapchat files with a hardcoded key implies that Snapchat is using a static key instead of using a different one for each message/between each user I believe.

- Yep I think this is the main weakness

=== Question 11 ===

What attacks is this protecting against?

- Man in the middle attack
- Phishing
- Smurf attack
- DOS

What attacks would not be protected by this?

- Employees leaking information - insiders
- Reverse shell attacks - random ppl sending tampered phone cables to them
- ??????? CANT THINK OF ANYTHING ELSE???
- OS or system 0-days

- anything that can spread via lan
  - e.g. Eternalblue
- physical attacks

- direct access to computers
- anything brought in physically, like on a flash drive
- could be deliberate or inadvertent

Give the 3 main factors in favour of Australia doing this

1. Reduces potential attack
2.

Give the 3 main factors in favour of Australia NOT doing this

1. Inconvenient / costly
2. Less responsive to outside
3. Internet used for research
4. Hard to communicate with outside bodies

=== Question 13a ===

What is the likely security purpose of including the date and time in the message?

- ??
- Easier to trace
- This information allows the computer to organize files chronologically. An incorrect clock renders these dates unreliable, making it harder to find files when you need them.
- Way of having an organized log
- Also makes it so that correct hases with odd time signatures (such as the 41st of the 14th month) can be identified
- Defend against replay attack where the attacker copies the message and send it many times

What is a fake deposit message which could be injected by his gang which has the same keyed MAC value as this intercepted message, but which states that Ned has deposited $90000 not $100

13062016090190000DEPOSiTTONEDKELLYJUNIOR 29

Was it this simple ? Just replacing the amount of 100 by 90000? Won't this give out a different keyed MAC code ?

^ I got:
13062016118990000DEPOSITTONEDKELLYJUNIOR 26

=== Question 13b ===

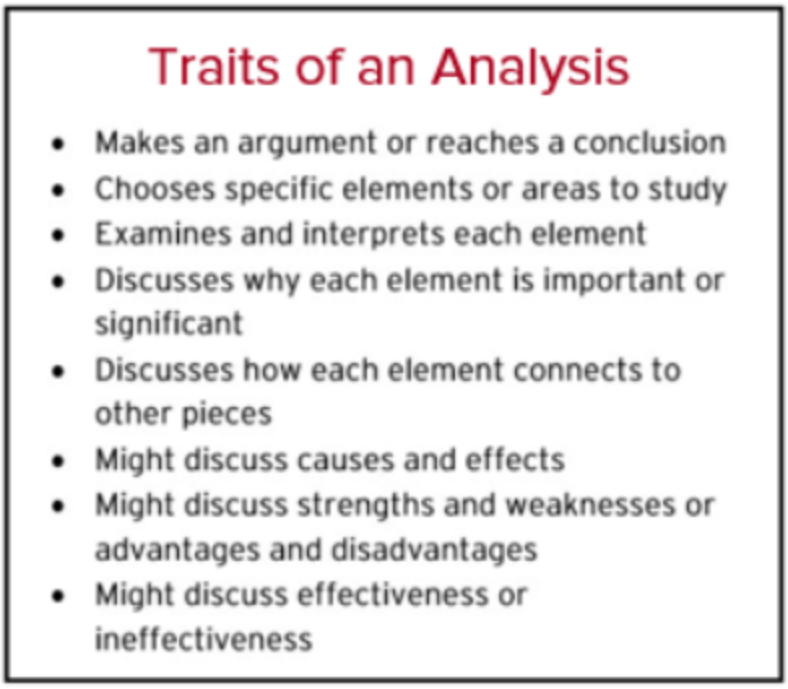
```
Decipher the message below to work out what to type into the answer field.

HQEET UTOII SNAEP MSRTT AOUIC DOINE

OEPOH YCNAE UDIHP RCETH TAIES EWINR

SCERS UNTGE IHUUT EFR
```

Anyone been able to solve this? - is there an online decoder of some sort that helps?
Cryptogram Solver (online tool)
Transposition Cipher Solver Helper ← I found this one useful too
OMG I Finally Got it (Here it is hidden)

HOW - It was kind of situational. Since the 2016 exam was something that they had to do by hand it couldn't have been a complex cipher. My initial thought was a transposition cipher just because of the layout. So working out the key was a combination of brute force and using the constraints of the English language. Seeing how there is a 'Q' and a 'U' they most likely be next to each other so then it was just finding patterns that could fulfill that then applying it to the rest of the cipher and see if an actual sentence could be produced. I hope this helps :)

Anyone know any good sentence rearrangers?

In case you didn't know what analysis is because they never said exactly what it meant my entire term



Case Study Summaries

Heya, could people please collate some summaries of the Case Studies we did? They shall be part of exams, but I've found that the best approach to case studies is having as many different perspectives as possible!

Thanks a ton you guys.

Case Study 1 - Reaction

Case Study 2 - Drill

- Notes by Sam O'Brien

A disaster on this scale given the number of safety fallbacks is very sad, it is too often that financial interests conflict with engineering requirements and businesses choose to take extreme risks for the sake of making money.

From the Interview

'Mud' is circulated to regulate pressure.

Cost 1 million dollars a day to have the rig there

21 days was expected for the oil to be reached

6 weeks (42 days) were needed

(1) A manager requested a faster pace, increasing the drill speed, this broke tools, removed the working fluid and broke equipment which meant the well was abandoned.

- At this point BP has already made a managerial decision that has cost the company 14 million dollars in time, broken equipment, logistics for moving the rig, lost working fluid but there has been no immediate disaster. ~25 mill lost

There was then an increase in pressure on crew, as again the idea was speed it up despite the fact that the same decision lead to a previous disaster only weeks before.e

(2) There was then an incident where a rubber seal (annulator) on a key safety device (blow out preventer) was damaged due to operator error. This occurred during a test and was caught when someone found chunks of the seal in the drilling fluid. When they brought it to the supervisor they were informed that it was not a big deal. Whether this was due to a lack of supervisor knowledge or a choice made by the supervisor due to pressures from management and the time to fix such an issue, certainly safety should come first.

(3) The blowout preventer had lost partial function of one of the control boxes weeks before which was not repaired. Admittedly, "some of its function" does not describe whether or not it needed to be replaced. If it had lost non-important functions, it may have made sense not to replace it, however if anything safety critical was damaged, relying on the backup is not an option, the backup is in case of an incident AND one failure of control, there is a second method of control.

They said they had tested it and it had passed, if this is the case, safety standards may need amending, or this may have been a non-issue.

(4) The different companies were discussing operations and had conflicting ideas about how to continue, creating a tense working environment and adding potential problems with operational function of the rig.

Mechanics of the Disaster

BP had issued their own report that had been criticised

All of the control takes place at the blowout preventer (at the bottom of the ocean) If there is a leak that enters the riser pipe, it cannot be stopped.

The vessel must constantly keep its self in the correct position (due to the drilling occuring directly below and the currents of the water moving the rig)

Crew consisted of 126 people in two shifts every 12 hours with a new crew every 21 days. (Hopefully communication was kept, as incidents should be exchanged between crews to keep people in the loop).

Difficult to access equipment, get replacement, whole new crew each shift.

If the rig was to deviate too far from the correct position, blowout preventor needs to separate the riser pipe, containing the drill. To do that, the blind shear ram essentially splits the drill and closes over the well preventing leakage of oils. Everything done on the sea floor is done remotely.

Out of two potential safety devices, the company chose the cheapest which had 2 barriers instead of 3. This choice its self is not inherently bad, however companies are always going to go for the cheaper choice.

Another safety device, a lockdown sleeve, was not installed despite the fact that they were ready to install one.

There was at the time NO standard for the conducting of a negative pressure test for determining whether the well is safe before continuing with operations. The test could have been conducted too soon, and was interpreted as safe when in fact was judged not by a secon opinion. This could have been due to time pressures once again.

The faulty negative pressure test indicated that the well was sealed, and in moving components, they did not monitor the fluid volumes which indicate the movement of oil.

They had to choose between diverting oil over the side or up to the rig and they chose the rig.

Electricity spiked from gasses entering generators, which caused electronics to explode, creating an ignition source.

The blowout preventer was closed early but due to prior incidents, did not seal the well.

The shear ram could be manually activcated but due to the power outage could not. In the event of a power outage as had occured the blowout preventor should automatically activate but it did not. In either case a remote operated vehicle can also activate the shear ram but again, it did not work.

Four main points:

Cement plug failure - Should have been tested properly and done correctly

Seal failure in the blowout preventer - Crew could have seen this if they were monitoring the fluids however they had confidence in the cement

Power outage caused by the intake of gasses into the generators overloading protections causing a loss in power which meant that the rig drifted.

Blowout preventer could not close the well.

Cementing the well has to be a pressure balance, due to the depth the pressure gradient is immense, over pressure was possible which fractures foundations and causes a leakage of cement, preventing a seal.

They chose not to perform a check to see if the seal was sufficient.

They didn't have sufficient stabalisers for the cement installment (6/21) they had the other 15 available but didn't install them.

Apparently the oil managed to go through 190 feet of cement related to the overpressure (3000 PSI)

They tried to get rid of excess lost circulation fluid which caused problems for the pressure tests when conducted on the well cap.

They had a difference of 1400 psi that should have been 0.

There are no automatic safety devices to prevent gas intake into the generators. Gas alarms were blaring but human intervention is needed to shut off generators. Automatic governers on the generators should prevent overspeed but they didn't stop gas intake and didn't work.

Loss of power means a drift which pulls the pipe up the blowout preventer which means damage to the blowout preventer due to tool join sections which are wider.

Out of 7 total ways of closing the well, 6 only worked if the annular seal worked. so the remaining way was the blind shear ram which failed.

Of the two control boxes, one had a flat battery, the other had a poor solenoid valve and ultimately it did not work. ROVs can provide pressure to operate it but the blowout preventer leaked.

The shear ram can't cut the tool sections and so had to line up well to work which it did not.

Shear rams should be tested both on land and when installed, ROVs which can operate them should be on site. the ROV on site was not compatable with the blowout preventer

Another suggestion was a status indicator on the preventer as there isn't feedback for it.

Media Coverage

The impact was worse than initially predicted, satelite imagery is only useful if the oil is visible which it isn't immediately or when in small distributed quantities.

Blame was passed between companies and settlements for damages were drawn up. Regulations were brought in to prevent additional disasters but as with all things, memories are short and a lack of accidents is often seen as a reason for less regulation. Under the Trump administration, a further removal of protective regulation was undertaken.

Removing 20% of the protections in order to expand drilling operations seems like a poor idea to me, but I am one for safety before progress.

Case Study 3 - Doors

Case Study 4 - Witness

Case Study 5 - Snoop

- Notes by Sam O'Brien

The old 'Nothing to hide, nothing to fear' argument

The first article (Encryption-busting laws)

Article has a clear bias, very clearly in favour of privacy over protections bill

Labor had opposed a bill called the Assistance and Access Bill which would allow Australian governments to issue three kinds of notice

Technical Assistance Notice TAN: compulsory notice for a communications provider to use an interception capability that they have

Technical Capability Notice TCN: compulsory notice for a communication provider to build a new interception capability to meet TANs

Technical Assistance Request TAR: This is arguably the most controvertial

While TAN and TCN are well defined and limited in scope, TAR is not, and won't be reported. An argument made by Dr Chris Culnane is that these have no oversight and are unlimited virtually in what can be asked of a telecommunications provider. They point out that there is not limitation on a TAR that they couldn't ask for a systemic weakness to be integrated.

They also argue that a communications provider could just be a single person who talks to at least 2 people, and basically all websites.

The defence minister tweeted "Labor has chosen to allow terrorists and paedophiles to continue their evil work in order to engage in point scoring," which is objectively a disgusting comment to make which aims to promote a particular stance.

https://www.zdnet.com/article/australia-now-has-encryption-busting-laws-as-labor-capitulates/

Mobile detection cameras

NSW has integrated mobile phone cameras to detect and fine individuals who use their mobile devices while driving, this uses AI to categorise images into drivers using their phone and drivers who are not. The positives are then checked by a person to confirm.

The idea is to reach a low number of road fatalities by cutting out a key distraction and providing a method of gathering data on drivers which is undeniable (nearly)

There is concern that this would flood the courts as with 1.8% of drivers caught with their phones, if 3% ended in court then thered be near 73 thousand cases.

In the time since it was implemented (Dec 2019), the amount of people caught has skyrocketed and as more cameras have rolled out, the numbers are undeniably larger with a revenue of $60 million from fines.

The below graph shows the effect of removing the warning signs before speed cameras, which may be implemented for mobile phone cameras, possibly resulting in a further increase

https://www.theguardian.com/world/2019/dec/01/world-first-mobile-phone-detection-cameras-rolled-out-in-australia

https://www.9news.com.au/national/speed-camera-and-mobile-phone-use-fines-spike-in-nsw/1686f126-6213-44ff-b1e7-1856bb3d1e0a

---

A&A, TOLA Act

While not explicitly stating that communications companies must break encryption but instead they may need to build functionality for spying to take place. Many point out that this capability means that it is much easier for a third party to access private messages and confidential data.

Specifically the government cannot ask for a systemic weakness to be implemented but the definition is ill defined. The company may be able to conceal that they have done anything and send an update notification to an app that provides a key security vulnerability to the software allowing keylogging or take screenshots of the users device.

The governement says that the targets of these laws are criminals, terrorists and paedophiles, but with all things, the government is corruptable and the question of how much power is too much power needs to be raised.

By negotiation with labor, the powers were limited to investigations of terrorism, child sexual offences and offences punishable by a term of 3 years or more in prison.

The Australian Human Rights Commissioner, Edward Santow, said Australia had "passed more counter-terrorism and national security legislation than any other liberal democracy since 2001".

One of those bills – the Espionage and Foreign Interference Act in 2018 – makes it unlawful for a current or former public servant to communicate information that "is likely to cause harm to Australia's interests" – including its foreign or economic relations. The offence can be punished by seven years in prison.

That act also contains an offence of "communicating and dealing with information by non-commonwealth officers" with a five-year prison sentence. So it could be journalists and whistleblowers, not just paedophiles, in the frame.

A targeted legislation to allow whistleblowers to be sentenced sounds like that was an aim all along.

Santow suggested that if the public became aware that law enforcement agencies could push an update of WhatsApp, for example, at one targeted user "it might discourage people from downloading security updates".

People point out things such as the NSA loss of equipment which ended up allowing the creation of the wanacry virus

Some definitions:

systemic vulnerability means a vulnerability that affects a whole class of technology, but does not include a vulnerability that is selectively introduced to one or more target technologies that are connected with a particular person. For this purpose, it is immaterial whether the person can be identified.

systemic weakness means a weakness that affects a whole class of technology, but does not include a weakness that is selectively introduced to one or more target technologies that are connected with a particular person. For this purpose, it is immaterial whether the person can be identified.

https://www.theguardian.com/technology/2018/dec/08/australias-war-on-encryption-the-sweeping-new-powers-rushed-into-law

https://www.legislation.gov.au/Details/C2018A00148

2020 TOLA Review by the INSLM

Suggested amendments:

- Give integrity agencies the ability to see TAR/TCR/TAN requests
- Remove some powers from civil immunity to ASIO
- Prevent agency heads from issuing TANs and the Attorney General from aproving TCNs instead give this power to the Administrative Appeals Tribunal
- Amend definitions such as systemic weakness
- Create a new office the Investigatory Powers Commissioner which should be a retired judge with technical advice who should assist in approving TANs and TCNs
- Non-Government requests should be approved by a tribunal

In previous reports, I have noted that the level of threat of a terrorist act occurring in Australia remains at 'probable', and the evidence I have considered for the present review indicates that this position remains unchanged. This review has caused me to consider broader security and other threats to the political, commercial and societal interests of the nation. There are real threats of foreign interference in facets of our lives that we may take for granted.

<u>Summary of recommendations</u>

Allow state and territory anti-corruption commissions to be given power to apply for the assistance notices

Move powers to tribunal

Creation of a new office to monitor and administrate.

Recommend that industry should be consulted

Increase the '3 year' requirement to be "Serious Australian Offence" and "Serious Foreign Offence"

change DCP to not refer to natural persons

Remove the AFP from involvement with the industry assistance notices

Confine the scope of civil liability

Explicitly state that ASIO cannot detain a person who is the subject of a request unless there is a lawful reason to do so.

Records should be kept of the number of industry assistance orders executed

Where non-sensitive information is concerned, make an open record of the requests

Do not allow ministers to remove material from an ombudsman report

Recommendation 32: "I recommend that there is no need to keep a record of IAOs that are not executed" Personally I disagree

https://www.inslm.gov.au/sites/default/files/2020-07/INSLM_Review_TOLA_related_matters.pdf

---

Facial recognition to replace opal cards

HEAVY DISAGREE

NSW government has suggested that there should be the ability to board public transport using only a facial image and not an opal card.

"I want people to not think about their travel. To quite literally turn up and go" - I think regardless some degree of planning is needed to execute this. You wouldn't just turn up to your house without a key and expect to get in, nor would you turn up to a train station without some indication of where you are going. Bringing a small card is a small price to pay for foresight.

Theres always the problem of False scans, or of mistaken identity. Facial recognition is statistics and as such wont work in all cases (i.e. twins).

Proposed method is an opt in method, supposed to speed up travel. Facial recognition does require quite a bit of processing and data base search so for safety, it would not be quicker than scanning a card. This idea is in my opinion formed from the nieve view that technology is magic and can solve every problem.

Another issue is the database of facial data needed to make this work. Concerns exist with the abuse of such a database, could it be used to target individuals unfairly, or be used for commercial uses by private companies in unethical ways?

People have a right to a level of privacy

An opt in or opt out system is difficult to integrate if all people are scanned regardless of their preference. If that doesn't happen, then there is a significant delay since scanning must occur only when requested.

https://www.theguardian.com/australia-news/2019/jul/11/nsw-suggests-facial-recognition-could-replace-opal-cards-in-not-too-distant-future

---

AFR discusses facial recognition

Since I don't have a subscription I didn't read this one

Australian views on surveillance

Factors affecting the perception on surveillance

- Is it neccecary
    - Many who said it was ok thought of it as very important
    - Terrorist attacks are the example used to argue for its use
- Do I trust the government
    - Current trust in the government is very low
    - People don't trust the government with the management of data

Error rate of automated technologies is a concerning factor that may turn up many false positives. No opt out system currently exists for surveillance data.

Law council of australia - *"It is unacceptable to assume the majority of Australians, who are not criminals and have the expectation to be kept safe by the state, are willing to succumb to heightened surveillance. "*

https://theconversation.com/australians-accept-government-surveillance-for-now-110789

Facial recognition used by Aus

The Capability is the name given to a database of information on everyone. Data from agencies such as the Roads and Maritime Services are to release information to a federal system.

Currently a Face Identification Service exists which performs a one to many match of an unknown person to help identify them. Access to this is supposed to be limited.

Criminal Juriprudence Professor Liz Campbell states that this is a breach of privacy rights as it allows the collection and storage of personal details of innocent and non-suspected individuals.

Supposedly in the pilot scheme 91% of matches were incorrect and targeted innocent members of the public.

ACT and VIC have objected to the program.

There is not a proper definition of how the data will be used.

Type 1 Type 2

https://www.smh.com.au/national/nsw/surveillance-state-nsw-intensifies-citizen-tracking-20181019-p50atw.html

---

The effectiveness of surveillance technology

Law enforcement these days now performs intelligence work to stop terrorist attacks before an event occurs. Currently intelligence agencies inform policy.

Intelligence is not in the business of looking for bad actors, instead it is looking at all peoples and forming a view.

Tactical intelligence relates to a defined operation while strategic intelligence is general and seeks to find the unknown. The goal of strategic intelligence is information dominance.

It is difficult to define the efectiveness of a strategic operation

Effectiveness metrics

Thwarted Attacks - Controvercial

Lives Saved

Terrorist organisations destroyed

Output

Context

Support

Informed Policy Maker

The effectiveness is usually from an organisation as a whole and not specific to surveillance technology

Mass surveilance is firmly not what intelligence officials conduct according to intelligence officials.

Intelligence organisations are firm about what they collect on their own citizens but are much softer on what they collect on other nations. Intelligence communities mean that these organisations share data on eachothers citizens therefore complying with domestic policy but obtaining the same outcome.

Brittain argues that to enjoy privacy you must have security and this balance is something that goes hand in hand. Public perception is very important as if a public is distrustful then the data collected will be useless.

the US considers this a balance issue while Brittain believes that they do not effect one another and can be done simultaneously.

https://www.tandfonline.com/doi/full/10.1080/01972243.2017.1414721

Australian Metadata Collection Laws

Telecommunications must store

- Incoming and outgoing telephone caller identification
- Date, time and duration of a phone call
- Location of the device from which phone call was made
- Unique identifier number assigned to a particular mobile phone of the phones involved in each particular phone call
- The email address from which an email is sent
- The time, date and recipients of emails
- The size of any attachment sent with emails and their file formats
- Account details held by the internet service provider (ISP) such as whether or not the account is active or suspended.[3]

22 Agencies can view this data without a warrant and data is retained for a two year period.

Overseen by a comission

Case Study 7 - Problem

Apollo 13 Summary

https://en.wikipedia.org/wiki/Telecommunications_(Interception_and_Access)_Amendment_(Data_Retention)_Act_2015

- mentions apollo 1 practice launch that killed its passengers
    - Jim Lovelle/Tom Hanks said that they fixed the problems with it (door)
    - His wife has a nightmare about the mission going wrong
- Ken gets sick with the measles - determined from blood test
    - This breaks up the crew and they have to go with someone else as substitute
    - The well oiled team that has been training for ages suddenly gets change 2 days before the lift off
- Simulations with substitute doesn't go well
- Engine 5 dies - as long as they burn the others should be ok
- Fred starts throwing up
- Jack stirs oxygen
    - Something goes wrong - wires come loose?
    - Multiple failures occur - at least 4
        - Immediately think it's an instrumental failure rather than the slim chance of 4 things all failing
    - Losing oxygen
- Systematically going through what could have gone wrong
- Turn off power and close off fuel
    - kills mission to the moon
    - saves data and does manual calculations - verified by several different people
- Earth crew were arguing about what to do
    - Least problematic option but takes too long vs
    - Other solutions that are unexplored but will get them to earth faster
- Talking to literally everyone involved in the building of the shuttle
    - Trying to save as much power as possible
    - Running through sequence of initiation to find out what are the most essential systems and how they can be turned on efficiently.
- Getting earth crew astronauts to do landing simulations to be able to figure out how to do the reentry
- Create carbon dioxide filter using random stuff on board

    - They are first making it on earth and detailing everything they do
    - Then they recall it to the astronauts to copy the procedure
- Never know what sequence of events will occur that will lead to a situation - Jim
    - when he was talking about algae leading him home
    - but he could only see the algae when his lights died
- Added extra sticky notes over buttons that would detach, leaving behind the other astronauts
    - double safety
- This mission was a successful failure
- ANYONE KNOW THE NAME OF THE SPACESHIP THEY WERE IN THE MAJORITY OF THE TIME? Aquarius

Case Study 8 - Ghost

What would you as the major Major do to get from Daniel his report on what to do about the Crystal Skull?

Toshihiro Tabata 26 days ago

What would you as the major Major do to get from Daniel his report on what to do about the Crystal Skull?

I honestly thought I was having a stroke reading this.

Reply You, Nina Yang, Akhil Kumar and 35 others like this Unlike

fax

Additional information about GDPR??

Good luck Everyone. One Good Term.

Good luck!

We got this!

Its gonna be easy dw