Protocols - Integrity

- Kahoot questions:
    - Individual letter frequencies
    - Bits of work
    - Identifying a lock picking tool
    - Risks
    - Substitution ciphers
- *Umbrella anecdote* - umbrellas make a good defence against cameras
- *Substitution cipher example on English alphabet*
    - fac(26) combinations
    - 26! is very big (~ 4 x 10^{26})
    - 100 x $(10^3)^8$, 100~7bits, 10^3~10 bits, total **~90 bits of work**
    - 16GHz, 16 x 10^9, 2^34(operations per second) * 2^12(seconds in hour) * 2^5(hours in day) * 2^9(days in year) = 2^60 operations (60 bits) for a year of running
    - Note:
        - 1000 ≈ 1024 = 2^10
        - 1 million ≈ 2^20
        - 1 billion ≈ 2^30
- It seems like it would take *too long* for us to *brute force* a substitution cipher, but we haven't taken into account letter frequency.
    - Also, brute force is not the only way to break the cipher.
- English language has *patterns* so brute force time isn't necessarily the time it takes to crack a cipher, redundancy in English, tenses in a sentence match up
- **Entropy**: degree of *randomness*.
    - Complements patterns or order
    - Less patterns = higher entropy
    - English has a lack of entropy
- In English, there are 2^25 possible combinations for a 5 letter 'word', but only 2^13 valid words.
    - Each letter adds 2.5 bits on average.
- Passwords are more likely to use valid English words or phrases than a random string of characters
- *Brute force*: easiest way to calculate and need to find domain of work.
    - **Guided Bruteforce** = bruteforce with heuristics
- Knowledge of the language also factors into time to crack cipher
- *Telegraph example*
- Possible attacks
    - **Replay attack** - an attack on a security protocol using replay of messages from a different context into the intended (or original and expected) context, thereby fooling the honest participants into thinking they have successfully completed the protocol run

    - **Man-in-the-middle** attack - a man-in-the-middle attack (MITM) is an attack where the attacker secretly relays and possibly alters the communications between two parties who believe they are directly communicating with each other. One example of a MITM attack is active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker
- *Challenge response* - a family of protocols in which one party presents a question and another must provide a valid answer to be authenticated; we can't replay a message from one bank to another since the challenge may be different each time

Nonce - a random number that is used once, used to ensure that old communications cannot be reused in replay attacks.

Non-repudiation - unable to deny it was you who sent the message (it had to have been you who sent it)

<u>Social Engineering</u>

Intro

- Art of learning and lying
- Relies on human interactions
- Most common examples include phishing, ransomware, USB baiting etc.
- 95% of all attacks involve social engineering
- It takes about 146 days to detect a breach caused by social engineering

Life Cycle

1. Investigation - learn as much as possible about the victim
2. Hook - initiate the conversation with the target and build rapport
3. Play - obtain the information
4. Exit - leave the conversation without seeming suspicious

Social Engineering Vectors

- Pretexting - invented scenario to get information out from someone
- Baiting - taking advantage of curiosity or greed
- Quid Pro Quo - getting something for something
- Tailgating e.g. heavy box technique (stand in front of sliding doors with heavy boxes so people are inclined to help you and let you in)
- Phishing - digital invented scenario e.g. email to people to get their credit card details. Phone phishing is also easier now because we can automate voices

Security Questions

- Humans often struggle to produce secure and unique passwords
- Social media makes it easier to answer security questions
- Pseudo-private information - information is less private for friends and family
- Prevention and strategy
    - Always lie - consider a security question as another password

    - Use password managers and scrub your social media
    - Don't reuse security questions and answers

Principles of Persuasion

- People can be easily exploited using persuasion
- Reciprocity - when you do someone a favour, people are often obligated to return the favour
    - Can't make it look like a bribe
    - Increase the time delay between the initial gift and the later request
- Liking
    - If you can get someone to like you, it's much easier to influence them
    - Presentation, body language, establish rapport etc.
    - E.g. brands on Twitter are relying on humour to drive sales
- Social Cues
    - People believe in the social cues around them
- Authority
    - People often blindly follow authority figures
    - Can influence other people based on what you wear, what you say, how you act etc.
    - E.g. APEC 2007 Chaser's War - bought expensive cars and were able to get quite far into a secure area, even though they had fake ids

Other

- Dumpster Diving - can go through rubbish with gloves on and find lots of information
- Even after you burn confidential documents, you can still obtain some information from the ashes

<u>Guest: Matt O'Sullivan (Sydney Morning Herald) on GIPA and FOI requests</u>

- GIPA [Government Information (Public Access)] allows members of the public access to government documents (subject to certain restrictions eg: commercial in confidence, National security etc.)
  https://www.ipc.nsw.gov.au/information-access/information-access-resources-citizens/how-do-i-access-nsw-government-information
  https://www.oaic.gov.au/freedom-of-information/foi-resources/foi-fact-sheets/foi-fact-sheet-6-how-to-apply
- Seeking a schedule of documents can be really useful, so you know what exists, and what to go for. Ministerial briefing notes are excellent summaries.
- Wording of requests is critical. Can be limited by time, or by subject.
- If your request is denied, you can appeal to the privacy commissioner.

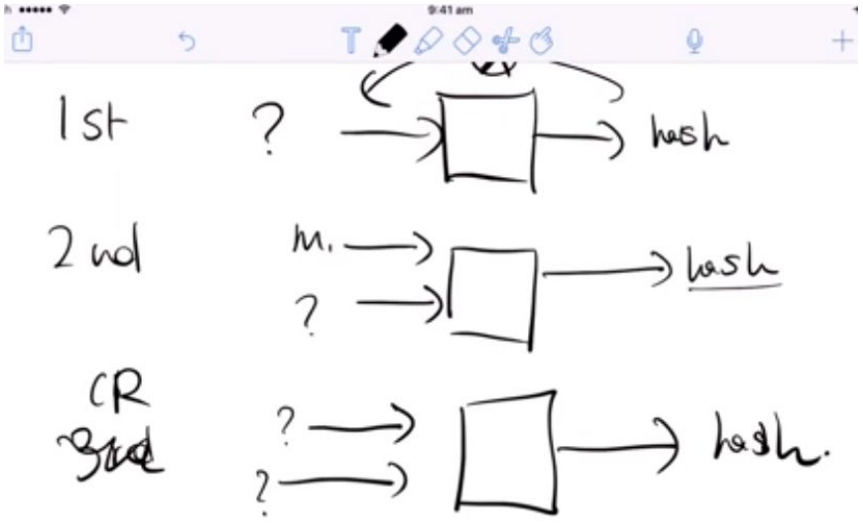
Hashing

- Hashing is a process that reduces any amount of text or data into a fixed size output
- If you can find collisions you can possibly defeat the hash
- Want to set the hash output to be the biggest it can possibly be i.e. even the square root (half the number of bits) is still too much work for an attacker to do

Desired resistance against the following attacks:

- preimage attack: given h(M) find M
- 2nd preimage resistant: given M find M' where h(M)=h(M')
- collision resistant: find any two messages M M' where h(M) = h(M')

- *Preimage* attack - giving the hash, find the message i.e. go **back to original** message from a hash; given *hash(x),* find *x*
    - Want hash to be preimage resistance - **Given** the **hash,** it should be **hard** to **find** the **original** message
    - Complexity on n-bit hash: 2^n. On average: 2^{n-1}.
- *2nd preimage* attack - **given** a **message** M and its hash, **find** another **message** with the **same hash** by going back from the hash; given *x* and *hash(x),* find *y* such that *hash(x) = hash(y)*
    - it should be hard to come up with a second message that has the same hash
    - Complexity on n-bit hash: 2^n. On average: 2^{n-1}.
    - More specialised version of collision attack
- *Collision attack* - **Attacker choose** any **two messages** where the **hash** is the **same;** find *x, y* such that *hash(x) = hash(y)*
    - Difference with 2nd preimage: attacker control 2 messages instead of 1
    - Want hash to be collision resistant - It should be hard to choose any two messages that result in the same hash
    - Complexity on n-bit hash: $2^{n/2}$. On average: $2^{n/2-1}$.



- Uses of hashing
    - Fingerprinting - check that the file matches the hash of the file. Quick summary that we can compare
    - Passwords
    - Proof of Work - need to do a lot of work to achieve something
        - E.g. bitcoin - you can only get a new block when you solve a puzzle, which takes a certain amount of bits of work
        - Hard to produce, easy to verify
        - Hashcash example: 20 first bits of SHA-160 must be zero
    - MAC (Message Authentication Code)

- Write a message, append a key, hash, send message and hash
- Receiver gets message, appends pre-shared key, and hashes to compare if message has been altered
- Provides I and A from CIA (since hashes already provide integrity)
- h(key | message)
    - Where key|message is key concatenated with the message
    - Vulnerable to length extension attack
- h(message | key)
    - Vulnerable if collisions occur
- h(key | message | key)
    - Same vulnerabilities as prev
- HMAC; h(key | h(key | message))
    - Better than MAC

Cryptographic Hashes

A cryptographic hash has some extra properties -

- Deterministic - The same text will always **result** in the **same hash**
- Its **quick** to **compute**
- It should be **hard** to **reverse**
- Collision resistant - It's **hard** to **find two** different **texts** with the **same hash**
- **Avalanche** property - A **small change** in **input** should **dramatically** change the **output** hash
- Varying input size to fixed size output

Symmetric & Asymmetric Cryptography

Symmetric:

- two people have the same key (shared secret)
- Every unique pair needs a unique key
- If there are 100 agents wanting to speak to each other, we need 99 + 98 + 97… keys (or (n^2 + n)/2 keys, which evaluates to 4950 keys.

Asymmetric:

- Public/private key pair
- Provides authentication, non-repudiation
- Encryption is very slow
- If there are 100 agents that want to speak to each other, we would need 100 private keys (1 private key for each agent)