COMP6441 Course Notes

KC Notes

http://kcnotes.wikispaces.com

Table of Contents

1. Introduction to Security Engineering

COMP6441 • KC Notes

1.1 Introduction to Security Engineering

- In civil engineering, bridges rarely fall down, people trust bridges.
    - Apply engineering skills and theories into computer security
- Consider differences between civil and security in terms of:
    - Attackers
    - Complexity and abstraction
    - Errors and controlling errors
    - Proving a bridge or program will work – or prove mathematically that it has problems?
- Think like an attacker
    - As a defender, you see only **the strengths** of a house, as an attacker you see **weaknesses**
    - **Question everything**, everything is vulnerable
    - Attackers can pick at one thing, defenders must defend all things.

1.2 History of Hacking

- History is very similar to the history of life.

- Phone phreaking – 2600Hz tone lets you call for free.
- User has **access to send control information** in the same channel
- Asleep at the wheel
- Like dinosaurs, not able to change after asteroid impact

↓

- Microsoft Money – now hacking **can get you money**
- Like suddenly being supplied oxygen

↓

- Specialisation: hackers can now coordinate attacks and specialise in one part of the process
- Gold hats: hackers find bugs and sell them
    - Bugs can be 'zero day exploits', where company does not know they exist yet

1.3 Targets

- <u>**Complexity**</u>: a complex system is difficult to secure
- <u>**Out of specification**</u>: when building software, you usually test for situations within the spec
    - Person looking for exploits look **out of the spec** and use systems in unintended ways
    - **WEP Insecurity**: initialisation vectors within packets
        - Again, control data was accessible (in this case, the destination address), and is accessible in inbound data

1.4 Case Study: Halifax Explosion

- Case studies are thinking problems – think analytically about how a problem can be fixed.
- Rather than **who to blame**, focus on **how to fix the problem**
    - Treat the problem as **a systemic failure** rather than an **isolated failure**

2. Security Literacy and Thinking

COMP6441 • KC Notes

2.1 Cyber Security Literacy

- **Reconnaissance** (recon) – gathering information and learning about a target
- **Active** recon: seeking information that **can be detected or identified** by the target
  - Engages with the target to get more information
- **Passive** recon: collecting information **without engaging** with the target
- The **little things mean a lot** – knowing the coffee shop nearby, overhearing phone calls can slowly open up to larger amounts of information
  - Other examples: dumpster diving, paper shredding, burning, hard drive disposal
  - All these can be reassembled and reconstructed

2.2 Think like an Engineer: Problems

- Hacking will be around forever. Why?
  - **Complexity** – misuse and abuse of unexpected and out of spec behaviour
    - The **Aneroid barometer task** – when asked to measure the height of a building "using an aneroid barometer", you could easily sell the barometer and buy a piece of string, or time and drop the barometer!
  - **Asymmetry** – attack one component, defend all components (Week 1)
  - **Weakest Link** – when one link breaks, the whole security breaks
  - **Hubris** – companies think like a defender (Week 1)
  - **Abuse of trust, human weakness** (Week 3), **psychology**
- **M&Ms** – companies believe that if you have one big protection system on the outside, everything inside is secure
  - That barrier is like an M&M – a brittle crusty layer that becomes **a single point of failure**
  - Difficulties in defining the boundary
  - Does not prevent attacks from inside
  - Examples: internal access points that bypass a firewall
- **Secrets** are similarly a single point of failure
- **Kerckhoff's Principle**: A cryptosystem should be secure **even if everything except the key is public**.
- **Security through obscurity** and **security theatre** – making things *seem* complicated and difficult

2.3 Think like an Engineer: Solutions

- To build a secure system:
    - Make sure the system is checked by others and yourself
    - **Ask the right questions** and figure out what you want to achieve first.
        - Sun Tsu: A poor general goes into battle and wants to win. A good general **plans how he is going to win without going to battle**.
        - Most systems have a lack of process and is cunning-less
    - Don't rely on obscurity as it is brittle.
- **Type I and Type II errors** – false positives and false negatives
    - When something is actually true but the **test says it is false**, or something is actually false but **the test says it is true**.
    - One of these situations is usually worse, but when we reduce the error of one we increase the error of the other. The best solution is **to reduce overall error**, but it is typically difficult.
    - Examples: making it easier/harder to get the dole/social services, jail, refugees entry/refusal, biometrics in airports

2.4 Crypto Literacy

- **Crypto literacy**: coming up with cryptography systems that work like magic
- **Protocols:**
    - **Confidentiality**: everyone can feel and see the system but only one person can do something with it, e.g. Japanese puzzle boxes
    - **Integrity**: messages cannot be tampered with
    - **Authentication**: how you know the message came from the owner
- **Primitives**: the building blocks to ensure CIA.

2.4 Confidentiality

- **Cipher**: a secret way of writing to ensure confidentiality
    - **Steganography**: hiding the fact that it is a message, e.g. tattooing a message onto a slave's head, pin pricks in a newspaper.
        - Easy to find, key can't be changed
    - **Substitution cipher**: replacing a letter with another (NSA app)
        - Frequency analysis and patterns
    - **Transposition cipher**: keeping the letters but rearranging their position (Rail fence)
        - Frequency of specific letters
    - **Old codes**
- **New codes** are judged by **entropy** – the amount of chaos in something
    - Considering pairs of letters – 26^2 possible pairs of letters
    - Bits of security – amount of work to brute force a cipher, usually best when the universe runs out (Week 3)

3. Risk and Key Cryptography

COMP6441 • KC Notes

3.1 Risk

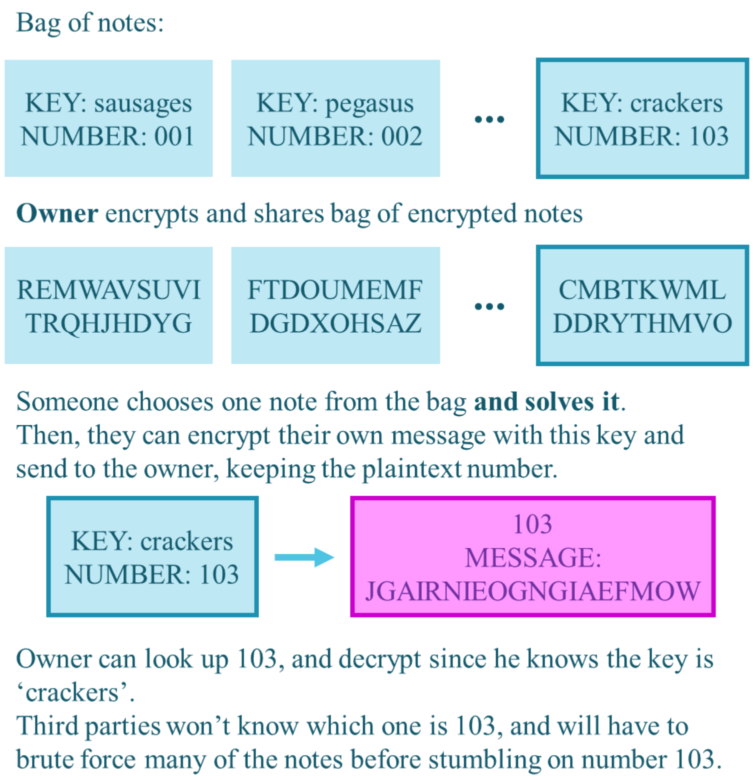
- Humans struggle to **weigh up risk and put a price on it**
    - Richard could repair his laptop in 2 hours or in 1 hour, but will not think the amount of care and attention behind the curtain – only **cares about the cost**
- We are **not rational** about risk
- You can usually <u>see</u> a bad event occurring, not the <u>chance of</u> a bad event occurring
    - Humans only focus on **the outcome**
    - People are just lucky or unlucky, but in the end **there is the same amount of risk** for each person

3.2 Low probability, high impact risks

- Humans are particularly bad at measuring risk when **the impact is catastrophic**, but it has **a very low probability** of occurring.
    - We usually shift this risk to another person (e.g. climate change)
- Examples: NASA gathering data on near-Earth objects, volcanoes, mass shootings, the San Andreas fault line
- We are getting good at measuring **risks in finance**, so we should apply ideas from finance
    - Both security and finance **protect something that might happen**
    - Little amount of past data in security, compared to finance
    - Companies struggle to know how much they need to invest in security
- In computing, low probability high impact risks are growing
    - **Companies storing everything in a central location** (single point of failure)
        - AWS, Google Drive, Gmail, Windows
    - This is a trade-off between benefits (speed, costs) and higher impact catastrophes

3.3 Keys: Ralph Merkle

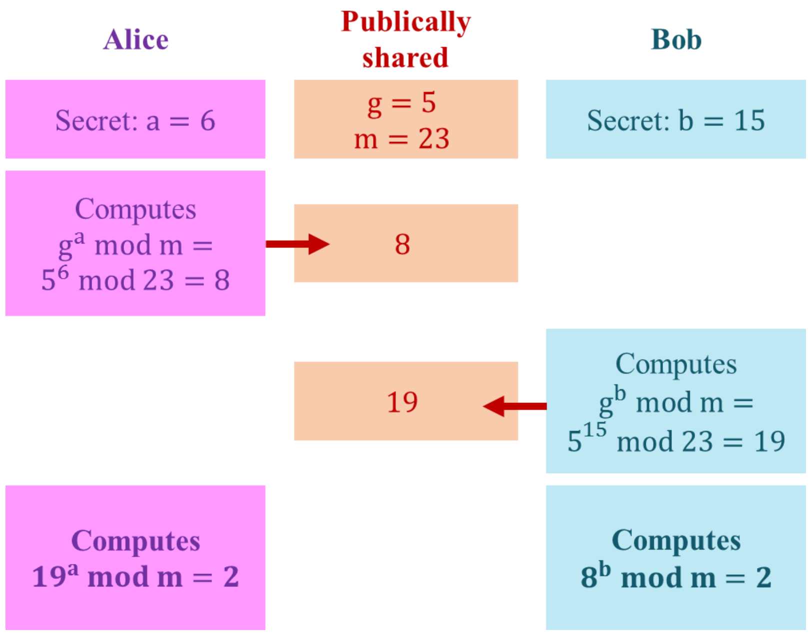
- With a large group of people, each pair of people need a secret key between them
- **Public key cryptography**: a public key for writing, a private key for reading messages
- **Ralph Merkle**:
    - Get a large bag of notes with different keys and a number. Encrypt both message and number, where the owner still knows the original message and number.
    - Share the encrypted messages.
    - Someone can **brute force and solve one note**, gaining one key and one number. They can then also encrypt, but **provide the number (e.g. 103) in plaintext**.
    - The owner can look up the number and decrypt using the key.
    - Third parties will have to **brute force and crack on average half the notes to find note number 103** in order to decrypt.


Bag of notes:
KEY: sausages NUMBER: 001
KEY: pegasus NUMBER: 002
…
KEY: crackers NUMBER: 103
Owner encrypts and shares bag of encrypted notes
REMWAVSUVI TRQHJHDYG
FTDOUMEMF DGDXOHSAZ
…
CMBTKWML DDRYTHMVO
Someone chooses one note from the bag and solves it.
Then, they can encrypt their own message with this key and send to the owner, keeping the plaintext number.
KEY: crackers NUMBER: 103
103 MESSAGE: JGAIRNIEOGNGIAEFMOW
Owner can look up 103, and decrypt since he knows the key is 'crackers'.
Third parties won't know which one is 103, and will have to brute force many of the notes before stumbling on number 103.

- Merkle's system relies on **asymmetry** – it takes much longer to decode every message than to solve one message.

3.4 Keys: Discrete Logarithm Problem and the Diffie-Hellman Key Exchange

- **Discrete Log Problem**: $g^a \bmod m = x$
    - Given g, m and x, **finding a is very difficult**.
    - We can utilise this difficulty to generate a private key
- **Diffie-Hellman Key Exchange**: how can you create a secret key between two people, without it being shared in public?
    1. Alice chooses a secret value **a**.
    2. Bob chooses a secret value **b**.
    3. Alice and Bob decide on public values **g and m**.
    4. Alice sends g^a mod m to Bob publicly.
    5. Bob sends g^b mod m to Alice publicly.
    6. Alice computes $(g^b \bmod m)^a \bmod m =$ $\mathbf{g^{ab}}$ **mod m using her secret value a.**
    7. Bob computes $(g^a \bmod m)^b \bmod m =$ $\mathbf{g^{ab}}$ **mod m using his secret value b.**
    8. Both now have a shared secret key, g^{ab} mod m!



Alice
Publically shared
Bob
Secret: a = 6
g = 5
m = 23
Secret: b = 15
Computes g^a mod m = 5^6 mod 23 = 8
8
19
Computes g^b mod m = 5^{15} mod 23 = 19
Computes 19^a mod m = 2
Computes 8^b mod m = 2

- It is hard to get a and b, even when given all publically shared numbers.

3.5 Bits of security

- **Bits of security**: a measure of the amount of work needed to decipher a key.
    - Measured in bits (base 2)
- E.g. if we have to do $1000 \approx 2^{10}$ operations (e.g. 1000 'attempts' to decrypt) then it has 10 bits of security
- Exploiting space/time trade-offs (using more than one computer) halves the number of bits
- 128 bits of security is a good ballpark for the universe to end

4. Human Weakness, Physical Security and Hashes

COMP6441 • KC Notes

4.1 Human Weakness: Problem

- **Humans are the weakest part to security**
    - **Greed**: corruption in police, bank tellers, abuse of trust and power
    - **Fear, emotion**: humans act and think irrationally
    - **Laziness**: humans do not like repetition, and routine checks may just be ticked off
    - **Pride, anger, curiosity, ignorance, overload of information**
    - Compounded by normalised behaviour – "this was always how it has been"
- **Costa Concordia** disaster and **South Korea's Sewol ferry** disaster
    - In the former, the captain of the boat left first and didn't think to evacuate the passengers first
    - In the later, the captain told everyone to stay on the boat
- Elaborate setups that are only **security theatre** – only *looks* secure
- Other disasters with systematic failures that need to be stopped, (e.g. child abuse, refugee and detention centre conditions)

4.2 Human Weakness: Response

- The response to human weakness is **training and drilling**
    - **Rick Rescoria** found the evacuation procedures for the World Trade Centre inadequate – trained and drilled his company's employees on evacuation
    - When the plane crashed, he evacuated and orchestrated the evacuation of around 2,000 people
- Similarly, training is needed for people to **act rationally when security is exploited**
    - Train people to stop tailgaters
    - Magicians and how they create distractions and trick you psychologically

4.3 Physical Security

- Having a secure communication protocol is useless **without protecting physical access**
    - Latest CIA leaks targeted physical access, e.g. televisions, optic fibres
    - Other physical access including stealing, key logging, microphones
- **Tamper-proof vs tamper-evident** – prevent tampering or know of tampering
    - E.g. ballot boxes with security tags need to be tamper-evident
    - ATMs need to be tamper-proof to prevent access to ports

4.4 Hashing

- **Hashing**: ensuring that a **message has integrity** (has not been tampered with) and this follows with **authentication** (message comes from owner)
  - Prevent a **man in the middle attack**, where someone could **change or replay** a message
  - Example: poker machine where a light beam reads the number of coins falling, but could be tampered with by covering up the light beam
- **Nonce**: a **number used once** that **prevents replay attacks**, e.g. the time of day
  - Time of day requires **confidentiality** – an alternative is a variable size or small fixed length appended to the string
- **Cryptographic hashing**:
  1. Sender and receiver decide on a secret, and sender appends secret to his message m.
  2. Sender hashes his plaintext secret and message and **sends the plaintext message m and hash h(m).**
  3. The receiver can **confirm by appending the hash to the plaintext and comparing hashes**.
  - Cryptographic hashes must be easy to go from m to h(m), but very difficult to go from h(m) to m
    - Passwords can be stored as hashes and you can compare hashes to verify user
- Attacks:
  - **Pre-image attack**: if given the hash h(m), you find the message m
  - **Birthday/collision attack**: if you find two messages m_1 and m_2 that have the **same hash h(m)**
    - The **birthday paradox** (not actually a paradox but counterintuitive) – as long as there are ~24 people, there is more than 50% chance that **at least two people share the same birthday** (number of pairs grow quadratically)
  - **Second pre-image attack**: given both the message m_1 and hash h(m), **you find an m_2 with the same h(m)**
    - Different from collision attack as **you are given more information**
- Current hashing algorithms:
  - MD5 – too small, easy to brute force. Not collision or pre-image resistant
  - SHA0, SHA1, SHA2 – all developed with the NSA, the first two considered **broken**
  - SHA3 not developed with the NSA
- **Broken**: once a hashing algorithm can be attacked faster than brute force.
- **Length Extension Attack**: because most hashes are iterative and take part by part to hash, so you could add to the end of a hash
  - **HMAC** solves this by applying **hash(k || hash(key || message))**

5. Vulnerabilities and Assets

COMP6441 • KC Notes

5.1 Vulnerabilities

- **Vulnerability**: a potential flaw or weakness
- **Bugs**: Human errors that are in the code
    - **Memory corruption bug**: changing contents of memory that were not expected to be changed
    - **Buffer overflow**: when there is a variable amount of space available and you write outside the buffer area
        - You can sometimes overflow and modify pointers (e.g. where a function will return to or go after it is done)
    - If you can *dream* it happening, it is true
- **Exploits**: taking advantage of a vulnerability
    - **Shell code**: do something that will pop up a remote shell or terminal
    - C programs are self-regulatory and usually check if someone isn't meant to be somewhere
- How do you know where to return to, to run malicious code you have entered?
    - Memory or information leak in the program
    - Brute forcing pointer addresses
    - Internal documentation
    - **nop sleds (no operation)** – you can fill the buffer with 0x90's, and you have a larger range to point to. It will then **slide down to your malicious code**.
        - **Intrusion detection systems** can detect nop sleds, but in the end this becomes **a cat and mouse game**
- **National Vulnerability Database**: a database of common vulnerabilities and exposures. Each one is a year and a number.

5.2 Assets

- What assets are we actually protecting?
- How important is each asset relative to each other?

5.2.1 STRATEGIES FOR IDENTIFYING ASSETS

- Involve **a lot of people** – people will bring their own takes and experiences
  - Dominant or arrogant people will lock others out of their views and shape an audit to their own ways
- **Develop a sensible plan**
  - It is easy to overlook things or centre on specific points – it may be good to brainstorm beforehand
  - Criticise everybody's ideas
- **Revise** the set of assets frequently – what they are and whether they are being protected

5.2.2 EXAMPLES OF ASSETS

- **Team America** – are we protecting against the terrorists or are protecting the innocent?
- **Car doorbell**: Richard's friend rigged a car door to a doorbell in their apartment, so that when the door opens, the doorbell rings.
  - Prioritising **the car or yourself as an asset**: is the car worth more than getting hurt from a criminal gang?
- **Coke**/other companies: reputation and branding is a very big asset
  - Difficult to put a price into a company's reputation

5.2.3 TYPES OF ASSETS

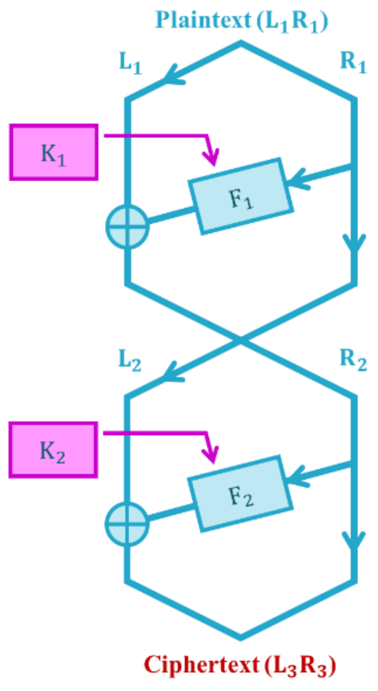
- **Tangible**: easy to value
- **Intangible**: hard to value (difficult does not mean don't do!)
  - Employee morale and security
  - Consumer information
  - Company secrets
  - Availability of services
  - Psychological and emotional

6. Ciphers and Stack Vulnerabilities

COMP6441 • KC Notes

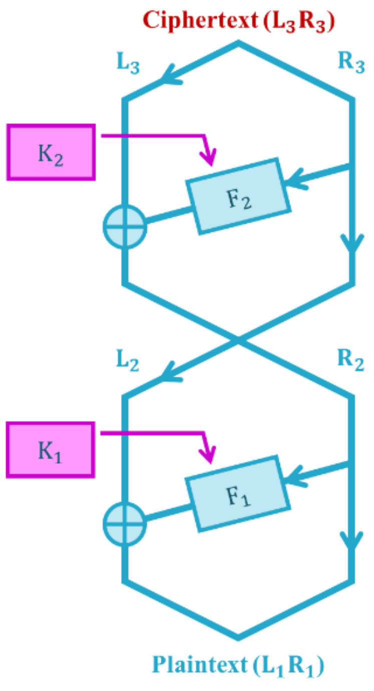
6.1 Ciphers

- Claude Shannon – a good cipher has two things:
    - **Confusion**: the relationship between the **key and ciphertext** is mysterious
    - **Diffusion**: the relationship between the **plaintext and the ciphertext** is mysterious
        - If someone changes one bit of the plaintext, we want at least half of the ciphertext to change
        - Most ciphers **avalanche**, so that it's hard to go backwards
- Dooke – looked at random numbers so that text could be properly encrypted
    - Something may **look chaotic, but plotted in 3D** can form regular lattice
    - Non-randomness helps make random numbers
        - Ciphers make **systematic changes** that are hard to undo
- **Lucifer**: a block cipher used by banks to transmit data
    - Was submitted as the Data Encryption Standard (DES)
    - Conjured up in a **competition by NIST** to submit good encryptions, run every five years
    - NSA changed numbers in the system (there's discipline and knowledge in Lucifer?) – they had made it resistant to differential cryptanalysis.
- **Fiestal**: **symmetric** structure where the plaintext is split into two – the first joined with a key and put into a function F, and then XORed with the other. This is **repeated** with another key.
    - Each round changes **half the bits** (the other half gets changed in the next round)



Encrypting	Decrypting
1. Split plaintext into two	1. Split plaintext into two
$L_2 = R_1$	$L_2 = R_3$
Join R_1 and K_1 and put into F_1 XOR function with L_1 $R_2 = L_1 \oplus F_1(R_1, K_1)$	Join R_1 and K_1 and put into F_1 XOR function with L_1 $R_2 = L_3 \oplus F_3(R_3, K_3)$
$L_3 = R_2$ $R_3 = L_2 \oplus F_2(R_2, K_2)$	$L_1 = R_2$ $R_1 = L_2 \oplus F_2(R_2, K_2)$

Note that encrypting and decrypting are nearly the same!
Also, because XORing twice results in the same string, your function can be anything (doesn't need to be invertible).



    - Because XOR is invertible, the function does not need to be invertible.
    - The same key is used to encrypt and decrypt
    - Relies on symmetric cryptography
    - **AES**: Advanced Encryption Standard works in a similar way with multiple functions.
        - AES-128 has 10 rounds, 192 with 12, 256 with 14
        - **SP network**: AES iterates between substitution and permutation ciphers.

6.2 Assymetric and Symmetric

- **Assymetric/Public key cryptography** is slow
    - It relies on **complexity in maths**
    - The public key may leak information about the private key
    - BUT it solves the key distribution problem
    - One way mathematical problems need to be **proven to be hard** to go backwards
- **Symmetric key cryptography** is fast
    - It relies on **complexity in randomness**
    - **Safes** are an example of a symmetric key: they are cheap and easy to encrypt, but they are certified to a time limit, e.g. 2 hours, 3 hours, rather than keeping it out
- **Side channel attack**: looking at metadata to attack a cipher/system
    - DES: can be attacked by monitoring power supply, latency, memory changes

6.3 Red Teaming (Guest Speaker: Finbar)

- Red team attacks, blue team defends
    - Red team is the enemy, company is being attacked, and the company is tested in how they respond to scenarios
    - Red team trains the blue team
- Focuses on the most important assets, and shares where it was easy or difficult to get through

6.4 Stack Vulnerabilities (Guest Speaker: Finbar)

- **Morris worm** (1988): one of the first worms distributed in the Internet, a buffer overflow
    - Independently discovered buffer overflow exploitation by Thomas Lopatic in 1995
    - 1996: PHRACK article, Smashing the Stack for Fun and Profit
- **Stack canaries**: a **tamper seal** before a return pointer – to overwrite the return pointer, you also have to overwrite the canary value, which is random
- **ASLR**: Address Space Layout Randomization, data within the memory space are shifted around
    - **PIE**: Position-independent executables, similar to ASLR
- **NX**: No-eXecute, to prevent code from being executed in the stack
- **Shadow stacks**: Hardware based isolated return pointer – it checks if the return address is the same as it was when function is called, before going back.

7. Top Men, RSA and Misdirection

COMP6441 • KC Notes

7.1 Top Men

- Security should never rely on “Top Men”
    - Brody: The Ark is a source of unspeakable power and it has to be researched!
      Maj. Eaton: And it will be, I assure you, Doctor Brody, Doctor Jones. We have top men working on it right now.
      Jones: Who?!
      Maj. Eaton: Top... men.
    - Everything needs **openness, scrutiny and oversight** in security
    - When people say ‘trust us’, you should never trust them.
- As a Chief Security Officer, reporting to CEO is better than reporting to Chief Information Officer, as they are the ones making the computer/IT system
    - Because CIO **sets up the system**, they **will always say it’s fine** (or sack you)
    - CEO is a better path to get to the board of directors – to get a decision/action
    - **Mandatory data breach notification**: compulsory notification that stock exchange will know about, and acts as a shortcut to the board of directors

7.2 Attacks and Security

- **<u>Side channel attack</u>**: attacks that are based on metadata of a system that is leaked from its use
    - An attacker can film a chip packet in a room with music playing in it, and transform it back into music
    - Attacker can listen to the CPU –1’s and 0’s may produce different sounds
    - Smartphone patterns easily visible when tilting the screen
    - Side channel attacks are **won’t be tested or proved** because it is hard to know about it.
- Attacks that **affect something later**
    - Terrorism before an event can change an election result
    - **Insider trading** before a **merger** allows an individual to purchase stocks in advance
    - Trading by saying a price will go down because of some threat
- Jewel heist: how can you improve its security?
    - Assets: people, jewellery
    - Threats, sorts of attacks: smash and grab, coercion
    - Mitigation: remove their ability to see – smoke, organic spray

7.3 RSA

- **RSA** relies on mathematical functions that are **hard to undo** and applied to crypto

1. Select a function, e.g. x^3
2. Select a set of characters we want to encrypt, e.g. [0…49]
3. Modulo the total number of characters in the set

- We require it to not have **clashes/collisions** (otherwise, it cannot be reversed with certainty)
  - E.g. A → 3, B → 3, but going backwards, 3 → A OR B
  - We can prevent clashes by **modding it with prime numbers**
- We can easily reverse RSA with a private key by **modding it with the product of two prime numbers (π)**, as long as both have nothing to do their primes less 1 $(\pi - 1)$

Encryption Function: x^k mod p

1. Let prime p be $7 \times 11 = 77$
2. Let the encryption key k be 7

You can share k = 7 and p = 77

x^7 mod 77

Example:
k = 3
$p_1 = 5$
$p_2 = 11$
Encrypt with x^3 mod 55 $-$ say x = 6
6^3 mod 55 $=$ **51**

Decryption Function: **y^n mod p**

1. Calculate m = $(7-1)(11-1) = 60$ from the prime numbers, **keep secret**!
2. Let n be the decrypt key where $n \times k$ mod m $= 1$ – here n = 43

Private: y^{43} mod 77

Alice tells us secretly $m = 4 \times 10 = 40$
k = 3
n is 27 (27×3 mod $40 = 1$)
Decrypt with y^{27} mod 55 $-$ y is 51
51^{27} mod 55 $=$ **6**

- Relies on the fact it is **easy to multiply two large primes**, but **hard to find a prime's factors**

7.4 Misdirection and Magic

- Magicians commonly use misdirection to perform magic, and **is similar to social engineering**
  - Make sure viewers **never concentrate on the important thing**
  - Exploit human weakness, greed

8. Identity and Authentication

COMP6441 • KC Notes

8.1 Intro to Authentication

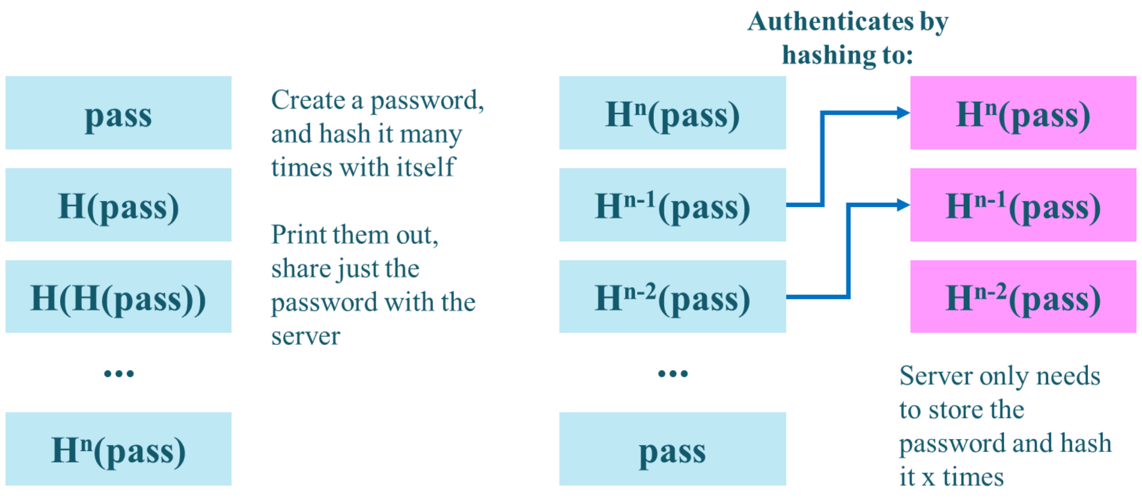
- When a computer program attempts to authenticate, their **only interaction is through 1's and 0's** – it is like looking only at a screen in front of you
  - **Authorisation**: whether someone has the permissions to access something
  - **Authentication**: verifying that someone is who they say they are
- Authentication can be done with **context or a pattern**
  - Must be used in plain sight
  - Is asymmetric – for example, a shared secret for each pair
    - Secrets **may not be tamper evident**
  - Should prevent replay attacks in case it is eavesdropped and altered
  - **Challenge response**: a way of authenticating you are speaking to the correct person
- The launch codes to US nuclear weapons were 00000000 in all locations
  - **Type 1/Type 2 errors** involved in launching
    - Either: having the military capable of immediately responding to a threat
    - Or: having another party launch the weapons easily, and relying on obscurity

8.2 Identity and Authentication

- **We authenticate with something we <u>know</u>, something we <u>have</u> and something we <u>are</u>.**
  - **Know**: a secret, but can be **leaked**
  - **Have**: a card, license or phone, but can be **forged**
    - A card also becomes **information** and becomes something you know
  - **Are**: fingerprint, way you walk
    - Fingerprints are **digitised and stored (knowledge)**. Also, it is difficult to change your fingerprints and can be lifted or read easily.
- **2 Factor Authentication**: if you compromise one factor, you may not be able to compromise the other.
  - E.g. Key logger can get your password (know) but not your 2FA app on phone (have)
- Identity theft **costs 2.2 billion every year**
  - Hard to change birthday, address or name
  - Driver's license: $500, Passport: $5,000

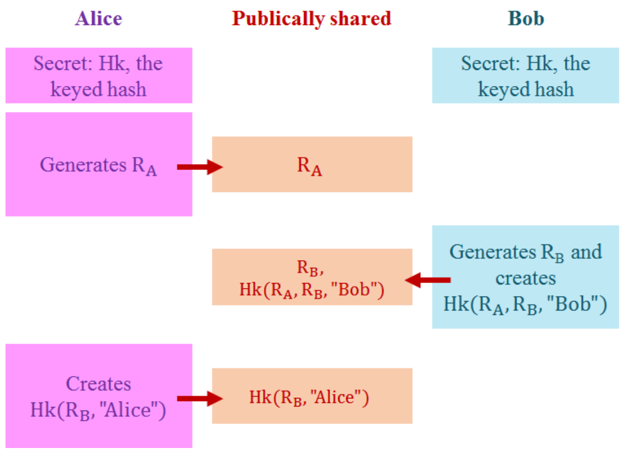
8.3.1 S/KEY

- **S/Key**: a password that is hashed many times – **prevents replay attacks**.
  - When we need to authenticate, we use the **second last hash** which will hash to the last hash. The next time we authenticate, we use the **third last hash**, etc.


Authenticates by hashing to:
pass
H(pass)
H(H(pass))
…
Hⁿ(pass)
Create a password, and hash it many times with itself
Print them out, share just the password with the server
Hⁿ(pass)
Hⁿ⁻¹(pass)
Hⁿ⁻²(pass)
…
pass
Hⁿ(pass)
Hⁿ⁻¹(pass)
Hⁿ⁻²(pass)
Server only needs to store the password and hash it x times

8.3.2 SKID ALGORITHM

- **Secret Key IDentification protocol** (SKID): a way of authentication that relies on a secret **keyed hash Hk()**.
  - Alice sends random number R_A
  - Bob sends back random number R_B, and the key-hashed string R_A + R_B + "Bob"
  - Alice sends back key-hashed string R_B + "Alice"



- We don't send just Hk(R_B, "Bob") to **prevent replay attack** (R_A is a nonce, just R_B could be easily replayed)
- We must send "Bob" **to prevent reflection attack** – we could be talking to ourselves without us knowing

8.4 Time of Check, Time of Use

- If there is a difference between **time of check and time of use**, it can cause a security vulnerability
    - If there is a time between checking if a user can **access a file**, and opening the file:
        - We can **change the filename before it opens but after it checks**
    - **Plane tickets** – they check your identity at security but at the plane, the boarding pass can be exchanged and you can board someone else’s flight

9. More Authentication and Trust

COMP6441 • KC Notes

9.1 Data, Control and Authentication

- **Data and control** is a very general idea and plays a large role in trust
    - Sending emails is considered **data**, and if abused, may **control your account** through links, attachments
    - **Authentication** of emails: susceptible to impersonation, fake website, man in the middle attack
- The problem is **to authenticate a website that people trust**
    - **Horton's Principle: "I meant what I said, and I said what I meant"**
    - We need to make sure **we are validating <u>the correct thing</u>**
    - What you think **what it means**, may not be the same as **what you see**

9.2 Two Authentication Approaches

1. **<u>Centralised Authority</u>**: a bank that gives you a signature, and you trust this bank
    - Based on command and control
    - Becomes a central point of failure
2. **<u>Decentralised, peer based Web of Trust</u>**: More people have their say and you trust these peers
    - Example: Pretty Good Privacy (PGP)

- Relies on **public key infrastructure (PKI)** – to talk to me, you only need a public key. To decrypt a message, you need a private key.
    - Man in the middle attack: **you encrypt with the wrong public key**, or person sends you the wrong public key
    - **X509** is the standard for public key certificates.

9.3 Certificate Authority (CA)

- Is an online and trusted
    - To sign via PKI, a company encrypts their certificate with **a private key** – people can **check this certificate with the public key**
    - Web browsers **come with preloaded public keys** from the CAs. All browsers do is **check if it matches**.
- However:
    - Anyone can become a CA, or **edit/create a certificate in your phone**
    - Why do CAs check and prove authentication? For **money**, e.g. people pay the cheapest CA, and CAs can pay to a browser
    - Malicious governments could place dodgy certificates and man-in-the-middle everything
    - **Horton's Principle**: you are authenticating **<u>the URL/email address</u>**, NOT the **actual company**. So, gitson-security.com may be a spoof of gitson.com, but a CA will authenticate it as owned by 'Gitson Security'.

9.4 Forward Secrecy

- **Perfect secrecy/Perfect Forward Secrecy**: if they learn your shared secret, **everything in the past is safe** and past messages and data is protected
    - This is usually not the case
    - E.g. S/Key: knowing the password means you know all the passwords

9.4.1 KEY DISTRIBUTION PROBLEM

- **<u>Key Distribution Problem</u>**: Distributing keys in public or over the internet is difficult as it must be done face to face. PKI solves this as a person can easily create a key pair and only share the public key.
- PKI is **more expensive** but solves the key distribution problem
- So, usually PKI is used to obtain a session key
    - A **slow, asymmetric** process, then a **fast symmetric** process for a period of time
    - Once a period of time is over, you can send another disposable public key
    - Obtaining a session key only unlocks a small part, **past data is protected**

10. WannaCry, Time and Knowledge

COMP6441 • KC Notes

10.1 WannaCry

- WannaCry was a recent malware affecting the world in a big way.
- **Incident response:**
  - **What have we learnt from this?**
  - **If you were IT, what would you have done?**
- Our suggestions:
  - **Disconnect, back up** your data, check if systems need to be live
  - **Assess your own system** – are the systems vulnerable?
    - At this time, **no one knows how to protect** and no one knows **what they are up against**
    - There are just theories, so you may just overreact to be safe
  - Tell staff members to not open attachments
    - Contact senior members
    - Internal response – if one system is compromised, the whole enterprise may be compromised
  - Have a **contingency system**, e.g. a paper based system next to the original system for ambulance services
    - It is a prudent act to have a backup or parallel system, in case one fails

10.1.1 RESPONSES IN GENERAL

- There was a planned choice to release the attack on **Friday**, when people are least alert
  - Attackers usually pick a time when people are **stressed, moving, closing**
  - It is when **defenders are least prepared and least ready**
- There is no need to panic – think and plan.
  - Weigh choices up, consider the cost of taking systems down
  - The Centre for Disease Control and Prevention (CDC) has many **contingency plans for disease outbreaks** – learn from cases outside cybersecurity
  - Ethics: make a decision that you won't be disappointed in later. Plan ahead and in advance

10.2 Effect of Time on Knowledge

- TOCTOU and Forward Secrecy (see 8.4, 9.4)
    - It is hard to prove something **has happened** **before a particular time**
    - It is hard to make a **tamper evident seal** on knowledge
    - It is also difficult to prove that **you have not leaked any information**
- **Proof of knowledge**: What protocols let us prove someone knows something, without sharing that knowledge?
    1. Post an encrypted file, and **later send the key**
        - Could possibly be decrypted by many keys to certain answers
        - Hash could reduce this possibility, though could be specially crafted
    2. Tell it to a **trusted third party or mutual friend**
        - Place it in a vault
    3. **Isolate** the information or the person who has the knowledge
    4. **Indestructible box with two keys**
        - Both parties must *simultaneously* open the box
    5. Demonstrate the power to know something on other similar objects
- **Zero Knowledge Protocol**: How I can convince you **that I possess some knowledge**, **without conveying that specific information**. In other words, to convey that I do indeed possess the knowledge, without sharing that knowledge.
    - **Prove that you can break into any house in the city**
        - Pick a city
        - The prover will build a replica house in the city
        - The verifier picks **where this real house is**, or **how to break into this replica house**
    - **Prove that an exam is easy**
        - The prover will write 30 questions into a hat, and
        - The verifier picks **20 questions from the hat**, and the remaining 10 will be in the exam
- Both examples have **two choices**, which **combined** **will leak the knowledge**, but when **separated and repeated** **will reduce the chance the prover is trying to trick us**.
    - This reduces the chance by ½ every time.

11. Incident Response, Privacy and Red Teaming

COMP6441 • KC Notes

11.1 Incident Response

- There are often capable people who respond to incidents.
    - People are affected by **emotions, stress, lack of information**
    - These people at the scene **should not be blamed** for the outcome
    - In the Lindt siege, there were uncertainties, anger from families and from the police
- Incidents like the Lindt siege do not go perfectly
    - In hindsight, there was bad planning right when the incident started
    - People should be **training** immediately for the event
    - **Plans** to be put in place
    - **Thinking** to be done in advance
- **Wisdom in hindsight**: we make systematic and thorough reports **after** an incident, but we tend to not carry out the information in future incidents
    - We should **learn from mistakes** from both in and outside of the cyber world

11.2 Privacy

11.2.1 OVERVIEW

- Secrets and information in the public usually **go one way** – they **spread**, and rarely can be **removed or put back** from the public
- **Project Angelfire**: plane with a camera can reconstruct what happens when an IED explodes
    - Plane flies there, **roll film backwards** to find the car and find who planted the bomb
    - Also works with robbery, assassinations – **roll the tape back, then roll the tape forward.**
- Benefits to information
    - Knowing more information helps police, security
    - Helps companies target products
- Risks to information
    - Protests: Chinese students gone missing
    - Women in abusive relationships tracked down
    - Other dissonant group information, like ethnic genocide
        - But can work the other way – driving Jewish people out of Denmark by looking at surnames in the phone book
- **"Nothing to hide"**
    - It may be easy to say it now when you trust the government, but **how about later**? The information would still be accessible and stays with the new government later.

11.2.2 WHO OWNS YOUR DATA?

- **Henrietta Lacks**: the owner of HeLa cells that helped **significantly in the medical field**
  - It is good that it is out there, but could be **abused**, and does not respect her privacy
- **Big Data**: primarily **personal data about individuals**
  - Who should control this data? Maybe governments, but companies?
  - Google NDA – the **intellectual property belonged to Google**, and ideas/IP had to be handed back
  - Two sides: **companies are glad to hold data** but **do not want to share it themselves**
- Individuals should **be more empowered**
  - Users should be able to have more control over their phones, and what sort of data it releases (e.g. MAC addresses, Wi-Fi access, EMF waves)
- **Perfect forward privacy**: where data from the past is safe

11.2.3 GUEST SPEAKER: NSW PRIVACY COMMISSIONER ELIZABETH COOMBS

- Privacy Commissioner:
  - Has no definitive power, but can ask for information, and has power of the Royal Commission
  - Is a regulator but **works with people** to prevent privacy breaches
  - Not a public servant, is independent – can **question government legislation**
- NSW Laws:
  - 1998 Privacy and Personal Information Protection Act (PPIPA)
  - 2002 Health Records and Information Privacy Act (HRIPA)
    - **Health information** is more sensitive and important
- Laws have no definition of privacy.
  - PIPPA: information used to reasonably **identify you**, including **images, fingerprints and an opinion about an individual**
  - HRIPA: personal (name, address) and physical functions
  - Not a black and white definition, and is **very contextual**
- There are separate laws for state and national levels – state can look at both public and private healthcare information
- **Privacy has transitioned from law based** to involving psychology, IT, engineering, maths
  - People are more aware of privacy and are asking more questions
  - People make assumptions until it starts affecting themselves – when they are in risk
  - Privacy **matters** when they are **a parent** (child info) or **getting a job**
  - No universal manners or etiquette for privacy on Facebook

11.3 Red Teaming (Guest Speakers: Westpac)

- **Red Teaming** involves testing **security controls** and **their effectiveness**
  - They **identify, improve and block** infrastructure and applications
  - Red Teams involve a lot of social engineering
- **Technology and protocols change**, but hacking has remained relatively the same
  - Is usually a cat and mouse game
- Red teaming involves:
  - **Diversity and community**
    - Different ways of interpreting and thinking outside the box
    - There are no hard rules, look through different angles
  - **Preparation**
    - People may unknowingly pass on emails, or call the police
    - Homework is very important, recon makes it become personal
    - Google is usually the first step in recon
  - **A Purple Team**
    - The team is an interface between the red and blue
    - Creates learning, uplifts monitoring
- Some tips:
  - Don’t click on dodgy links
  - Check your Facebook privacy settings
  - Make sure everything is patched

12. Whistleblowers and Bug Bounties

COMP6441 • KC Notes

12.1 Whistleblowers

- **Information easily flows, and it is difficult to keep a secret**
- Employees may internally tell the outside world when they are not meant to, when a company is not doing good things
    - American elections – when a candidate is bad
    - In security, we should consider not just why and what, but the **consequences** of leaking information
- Whistleblowing is **usually against a powerful organisation**
    - There is usually a standard way whistleblowers are treated:
        - Companies attempt to distract the media by discrediting the whistleblower
        - They are targeted – know that powerful people will ruin your life.
- Australia does **not have good whistleblower protection**
    - Be a Snowden
    - Be very careful if you do share company secrets
    - Cover your tracks

12.2 Vulnerability Disclosure and Bug Bounties (Guest Speaker)

- Vendors **do not need to respond to vulnerabilities**
    - People who find vulnerabilities threaten to sell, share or disclose a bug
    - St. Jude pacemakers: researchers found they could remotely drain battery and administer shocks
    - Often involves not only security, but the financial sector
- **Don't monetise security research you have already done**
    - If it ends up on a legal team, it may have consequences
    - Don't do stupid things, like DOS
    - Be practical: do bug bounties
    - Find a balance for when you stop when you find a vulnerability
- Bug bounties are hard to get into now
    - Find a niche – for example, the speakers found their niche in crawling for new Amazon services or new sites that were new and not documented