**Morning Lecture:**

**ANU hack - what would you advise UNSW Vice Chancellor?**

* Don't suggest hiring consultant
* Security engineering students to try and hack the system: penetration testing
* Normal answer - make it harder to break into uni with firewall
  + Supported by a design of the system so a break only gives you access to some things, store different things in separate locations
* Consider and analyse UNSW cyber email –
  + See Blog Post

**4 main concerns for seceng:** Trust, Secrets, Humans, Engineering (risk, complexity)

* Trust: be sceptical when you begin
  + Bad security is security that depends on trust
  + Who can you trust? No one; Who can you trust least? Yourself.
  + Physical examples: can’t trust locks, ciphers or fences, all these are breakable

**Defence in depth:**

* System should be able to work if something fails e.g. submarine division into compartments
* Avoid single point in failure, as was done with: Concentric rings of walls for castles - if one wall breaks, only access a small part of the land inside, with more walls between you and the central keep
* Segmentation of confidential data

**Security by design** – build security in from start, instead of in response to breaches

* You take people attempting malicious things as given and build system from the start with as few vulnerabilities as possible

**Complexity: Apes vs ants**

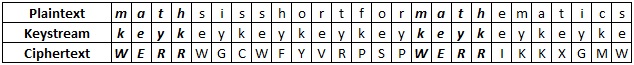
* Ants are fairly useless individually; it is the structure of the colony that allows them to achieve complexity
  + The colony is rigid, ants know and follow their roles
  + Individual ants die -> others fill their spot -> colony perpetuates
* Apes – complexity from individual agency
  + Ability to break rules - complexity comes from interactions betn individuals
* Brains develop to allow us to function in society
* The problem with ant and hierarchical structure is the single point of failure: what if the head of the structure is incompetent
  + E.x. Bell LaPadua – levels of confidentiality:
    - Not allowing people to access specific parts of info
    - Relies on complete human obedience
    - Can only increase controls + penalty to deter
      * Kennedy could have launched missiles whenever he wanted

**Physical security:** In our virtual space, everything we run is running on the machine – if someone interferes with the machine, we are screwed

* Best way to ‘hack’ a password: Physically – watch or film them typing it; look for grease and smudges on a phone for touch passwords
* Very difficult to physically destroy data -> we can reassemble shredded paper, etc.
* Side Channels - for every cyber action, there is a corresponding physical trace
* No software can guarantee protection from a hardware attack – i.e. the best way to prevent someone hacking your webcam is to cover it so they don’t see anything.
  + Click disable on you microphone: if someone’s already in -> does nothing
* Best way to stop physical attacks is usually through physical security

**Evening Lecture:**

**Vignere Cipher (Caesar + Password):**

* Take a password and shift the encrypted message by the alphabetical order of the password. e.g. ABBA would give the shift (0, 1, 1, 0) on repeat, so “Hello” would be:
  + H+0(A) = H E+1(B) = F L+1(B) = M L+0(A) = L O+0(A) = O [HFMLO]
  + Kasiski test: used to crack cyphers where offset repeats, like Vignere
    - Look for repeated strings in the cipher text – usually of length 3+ characters.
    - Example:
      * In the bottom row, there is a repeated string after 15 characters, which means the key length is a factor of 15, otherwise the key wouldn’t repeat in a space of 15 letters, it would be EYK or some other arrangement
      * So we know it’s either 3, 5 or 15 – assuming it’s not a Caesar
      * Intercept more messages and find that the key is 3 letters long
      * **Then group every 3rd letter (1, 4, 7, 10), (2, 5, 8), (3, 6, 9) and do a freq analysis on each chunk to find Caesar it**
  + **Coincidence Index:** easiest ways to determine if you have a monoalphabetic substitution cipher. The encoded message should have the same coincidence index as the language it was encoded in. (E.g. English texts typically have a coincidence index of 1.73)
* https://lh4.googleusercontent.com/tB1h7aXDQoFkQNWwvtMsaNM1OkqLwdAKp0_1sLq_T2oK2XBpF34aWINEkxLO4iFsHAi4LJNHAbhQI_I7d1-isY7wDmMVCQgpDZmF5qidjyUjymbmzBgbse7PLtmL2jbKmuBFrDof
  + - Formula: Where c = normalizing coefficient (26 for English), na = #of times the letter "a" appears in the text, and N is the length of the text.
  + **Manual Calculation:** Line the text up with a copy of itself shifted slightly.
    - Count the # of times the letter in the first copy matches 2nd copy.
    - Divide this by the number of aligned pairs of letters.
    - Multiply that by the size of the alphabet (26 for english)
    - The result is an estimate of the Coincidence Index

**Enigma Machine**: <https://www.youtube.com/watch?v=ASfAPOiq_eQ>

**One Time Pad:** generate a random sequence of numbers that is of equal length to the message you are sending, this is the offset used for each letter

* Then the receiver uses the number sequence to reverse the message
* +’s: gives uniform dist since randomness (26 sided die)
* -‘s: how do you pass the key securely & if you send a list of keys to use right from the start, each message must conform to the pre-determined length

**Type 1 & Type 2 Errors**:

* Type 1: False positive: E.g. a medical test says you have a disease when you don’t
* Type 2: False negative: E.g. a medical test says you don’t have a disease when you do