# COMP6441 Lecture 9: Identity and Authentication

Youhan Cheery

May 2017

- Authentication messages based on shared secrets allow the distinguishing of one person trying to access to another.
- Problem: the secret can (and likely will) eventually come out. Can also be susceptible to a relay attack.
- Maximum number of people to hold a secret is one.
- Challenge response is unreplayable, but still suffers from the authentication aspect.
- Authentication - something we know, something we have, something we are.
- Something you have that authenticates you is for example a driver's licence. Problem with license is that it is easily forgeable. This tends to be the case with things you have. *Homework: find things that are forgeable.*
- Something you are - fingerprints (DNA). Fingerprint is digitised and read, reader reads and digitises and the computer compares them. A fingerprint is just a piece of information. Don't have to remember it, since device remembers it for you. Problem: you can't change your fingerprint. So if it's ever compromised, there is nothing you can do. *Try to lift a fingerprint.*
- Two factor authentication: provides 2 layers of security as an attacker needs to break through 2 levels as opposed to just one. Problem is that with a lot of current technologies, there isn't much Independence between channels.
- Another problem with biometrics is that how much information do they actually store? The information is biologically limited and inflexible. Is it enough?
- Some simple authentication protocols:
  - Send secret, host compares (what about trust of channels)

  - Send secret, host stores hash
  - Salt the message and then hash
  - Send message encrypted with a secret (such as a nonce)
  - SKEY generates really large random number $n$. This then hashes the number with a cryptographic hash $h(n)$. Hash that to produce $h(h(n))$. Repeat this process $m$ many times. You input the initial one when you log in the first time. This hash is then removed and the next time you log in with the second one. Uses the fact that cryptographic hash functions are really hard to come back from.
  - SKID. Hk is keyed MAC, both know k:
    * Alice picks a random number, $R_a$ and sends this number to Bob.
    * Bob sends back to Alice $R_b$ which is a random number generated by Bob. Also sends back $Hk(R_a, R_b, Bname)$.

- Authentication vs authorisation: authentication is someone is who they say they are. Authorisation is the person who is applying is permitted to access data.

- Lag between checking and doing is a danger. Anything can happen in between.