Week 1

Welcome

THE UNIVERSITY OF SYDNEY

Acknowledgement of Country

I would like to acknowledge the Traditional Owners of Australia and recognise their continuing connection to land, water and culture. I am currently on the land of the Gadigal people and pay my respects to their Elders, past, present and emerging.

I further acknowledge the Traditional Owners of the country on which you are on and pay respects to their Elders, past, present and future.

Emergency procedures (on campus)

- In the unlikely event of an emergency, we may need to evacuate the building.
- If we need to evacuate, we will ask you to take your belongings and follow the green exit signs.
- We will move a safe distance from the building and maintain physical distancing whilst waiting until the emergency is over.
- In some circumstances, we might be asked to remain inside the building for our own safety. We call this a lockdown or shelter-in-place.
- More information is available at www.sydney.edu.au/emergency.

Health and safety advice

- The University is following NSW Government and NSW Health guidance to prevent the spread of COVID-19, respiratory-type illnesses and other illnesses.
- All staff, students and visitors are required to follow our health and safety advice. This includes staying at home if unwell, isolating and testing, and to not return unless recovered or as advised by your health professional.

Wellbeing and care

- If you become unwell during the semester, or need to stay at home, please notify your unit of study coordinator, as with any unexpected absence.
- If the illness impacts assessment, use the usual mechanisms including simple extensions and special consideration to arrange reasonable adjustments.
- Other helpful study information can be found on the student website.

Student support

- Visit the Student life, wellbeing and support webpage to find out about the student services, resources and events available to support you while you study:
  - Health and wellbeing
  - Academic support
  - Personal support
  - Getting connected

Questions about getting started this semester? Come, visit us at a Welcome Hub

Anderson Stuart
Welcome Hub

Carslaw West
Welcome Hub

Safer Communities Office

- Support and case management for people who have experienced sexual misconduct, domestic/family violence, bullying/harassment or issues relating to modern slavery.
- Contact the team
  - 8:30 am to 5:30 pm Monday to Friday, Sydney local time
  - phone: +61 2 8627 6808
  - email: safer-communities.officer@sydney.edu.au.
  - campus: Level 5, Jane Foss Russell Building, City Road, Darlington Campus
- Make a report
  - Visit the website to make a complaint or disclosure of sexual misconduct to the University.

Academic integrity

- Academic integrity refers to behaving honestly, ethically and responsibly in relation to all elements of your study at the university, including assessments.
- Always submit your own work, sit your own tests, and take your own examinations.
- Acknowledge any contributions in your assignment which are not your original thoughts, ideas or words. This includes contributions from assistive technologies (ChatGPT, Grammarly, etc.)
- Some courses may not allow the use of assistive technologies to create or modify work for submission, unless expressly permitted by the unit coordinator.
- Academic Honesty Education Module 2025 – all commencing students must complete by census date. Continuing students can self-enrol at any time.

Unit Overview and Assessments

THE UNIVERSITY OF SYDNEY

Teaching staff

- Unit Coordinator
  - Dr. Wibowo Hardjawana (wibowo.hardjawana@sydney.edu.au)
- Lecturer
  - Brian Udugama
    - Email: (brian.udugama@sydney.edu.au)
    - Consultation Times: Online consultations by email appointment
- Tutors
  - Avrati Motwani (amot0381@sydney.edu.au)
  - Quanhao Chen (qche7810@uni.sydney.edu.au)

Timetable

- Lecture
  - Friday - 3-5 pm
  - H04 Merewether Building. Merewether Lecture Theatre (2) 136
  - Attend in person (recommended); Recordings available on Canvas

- Tutorials\Labs
  - Monday or Tuesday - 12-2 pm
  - J15 Shepherd St Building. Computer Software Lab 101
  - Attend in person to work in groups (required)
  - Week 9 Tutorial Quiz and Week 13 Project Demo

Assessments

- Projects 35% - Group assessment
  - Written assignment and Python code to be submitted on Canvas
  - Project 1 (15%) due in Week 7
  - Project 2 (15%) due in Week 12
  - Project Demo (5%) in Week 13 during timetabled tutorial

- Tutorial Quiz 15% - Individual assessment
  - Online Canvas quiz in Week 8 during timetabled tutorial
  - Multiple choice and extended response

- Final Exam 50%
  - Supervised exam during formal exam period
  - Multiple choice and extended response

Unit overview

- Lectures
  - Provide theoretical foundations of computer and networking security
  - Tutorial quiz and final exam will be based on lecture materials

- Projects
  - Programming skills needed to understand and modify Python code
  - Initial labs in Weeks 2-4 will introduce fundamental security concepts required in projects.
  - Project 1 looks at cryptographic security for botnets. Focused on understanding key exchange, confidentiality, integrity, replay attack prevention, and authentication.
  - Project 2 looks at how to prevent common computer and network security attacks including source code and SQL vulnerabilities.

Unit schedule

Weeks	Lecture/Lab
1	Introduction
2-4	Hash functions, MAC, Ciphers, PRNG, Diffie-Hellman Key exchange, DES/AES, RSA **Lab 1-3**
5-7	Authentication, Cryptographic certificates, SSL/TLS, Cryptographic protocols, Network Protocols **Project 1 (15%) (due in Week 7)**
Easter Break (Week 8)	
9-12	**Tutorial quiz (15%) – week 9** Blockchains, Software Security, Network Security, Web Security, Wireless Security, Hardware Security **Project 2 (15%) (due in Week 12)**
13	Quantum Cryptography, Review **Project demo (5%)**
Exam Period	**Final supervised exam (50%)**

Additional resources

- Please email the lecturer and tutors with any problems
- Check announcements and assessment deadlines on Canvas
- Use discussion boards on Canvas for general queries
- Recommended programming and reading resources:
  - Grok Academy (groklearning.com)
  - William Stallings, Cryptography and Network Security: Principles and Practice (4th). Prentice Hall, 1999. 0130914290.
  - A. Menezes, P. Van Oorscho, S. Vanstone, Handbook of Applied Cryptography (5th). CRC Press, 0-8493-8523-7.

Introduction to Security Basic Concepts

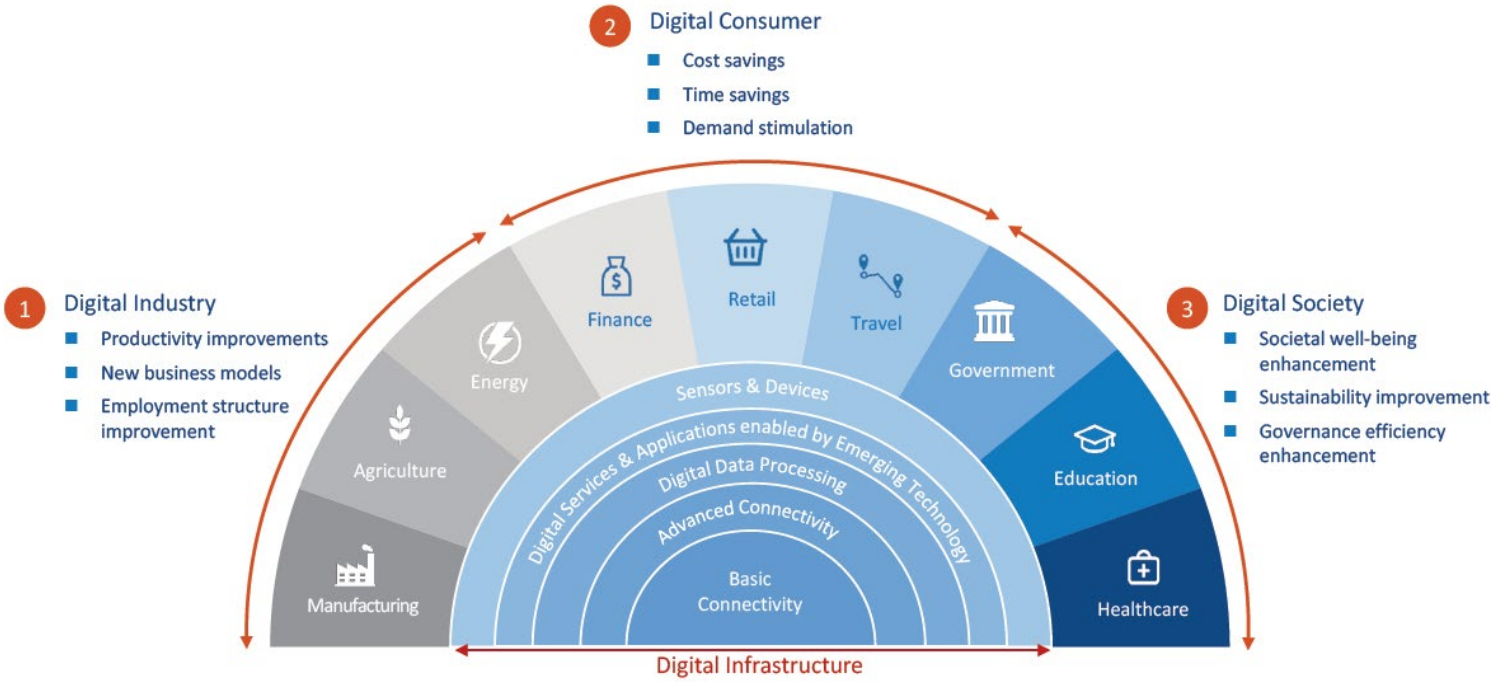
THE UNIVERSITY OF SYDNEY

Security headlines

- Economy
  - UN experts investigate 58 cyber attacks worth US$3 billion
    - "… cyberattacks on cryptocurrency-related companies between 2017 and 2023, valued at approximately US$3 billion …"
- Government
  - US government data breach linked to Atlassian bug
    - "… bug in enterprise software maker Atlassian's Confluence suite of collaboration tools …"
- Safety
  - Microsoft patches 80 vulnerabilities
    - "… exploitation would allow an attacker to bypass the Office Protected View and open in editing mode rather than protected mode."

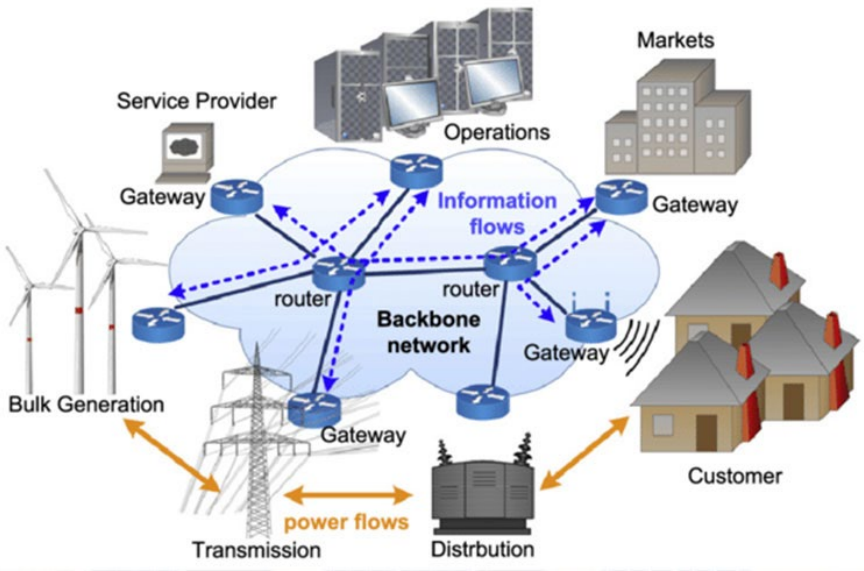
The world we live in

- Digital information age driven by global internet connectivity and high-speed wireless technologies
- Digital transformations have impacted critical infrastructure assets in various sectors including energy, banking, healthcare, food, water, and transport.



Source: Arthur D. Little analysis based on publicly available data

The Internet of everything

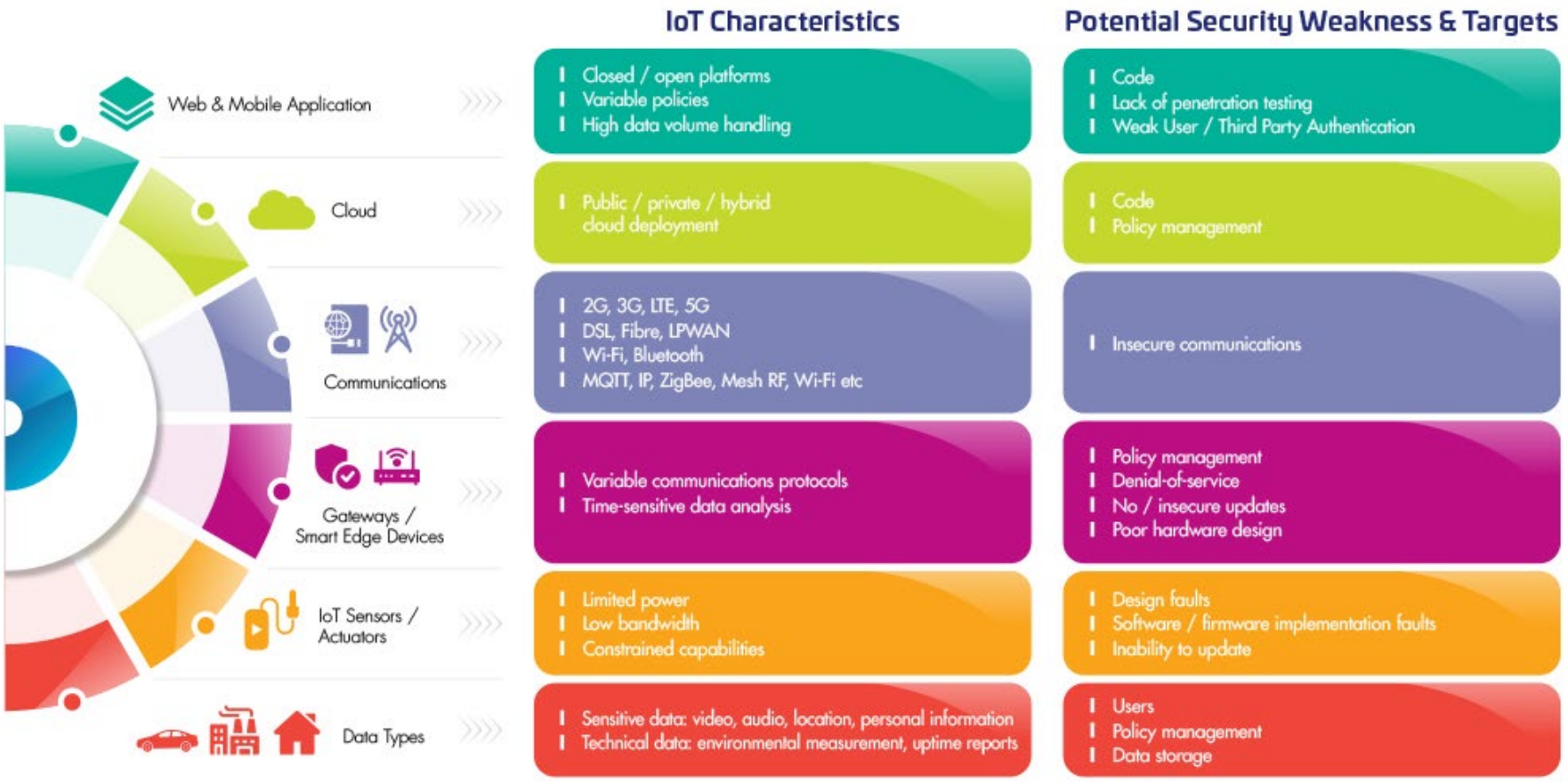


https://www.sageautomation.com/blog/australias-smart-grid-future-when-we-can-expect-change

Application Specific IoT (ASIoT)	Smart things and objects in ASIoT	Design challenges and optimization parameters
User domain-driven ASIoTs		
Internet of Battlefield Things (IoBT)	Sensors, munitions, weapons, vehicles, human-wearable devices	Fast adaptive robust network communications, real-time information processing, high security for IoBT functioning
Internet of Medical Things (IoMT)	Medical wearables (Parkinson, multiple sclerosis, diabetes, heart rate, ECG), smart things (insulin and inhalers)	Interoperability between manufacturers, simple connectivity and device management, security and privacy concerns
Internet of Animal Things (IoAT)	Smart cattle collars (rumination, activity temperature), RFID ear tags, sound analyzers	Energy efficiency for on-animal measuring devices, indoor wireless channel characterizations
Internet of Waste Things (IoWasteT)	Smart garbage bins (SGBs), RFID tags, cameras, actuators	Energy efficient large scale data collection, integration with IoV, data MULEs
Communications medium-driven ASIoTs		
Internet of Underwater Things (IoUWT)	Underwater sensors, smart buoys, AUVs, ships	Long propagation delays, high error rates, short bandwidth, difficulty to recharge devices
Internet of Underground Things (IoUGT)	Buried soil sensors, seismometers, mobile sinks on field vehicles/machinery	Power conservation, network topology design, antenna design, environmental extremes
Technology constraint-driven ASIoTs		
Internet of Nano Things (IoNT)	Nano-sensors and actuators, nano-routers, nano-micro interfaces, bioFETs	Nanomaterial properties (e.g. graphene nanoribbon (GNR), carbon nanotube (CNT) for communications
Internet of Mobile Things (IoMobT)	Mobile personal devices (tablets, smartphones), robots, vehicles	Dynamic network topology, unpredictability, direct energy exchange among smart objects

"Application Specific Internet of Things (ASIoTs): Taxonomy, Applications, Use Case and Future Directions", IEEE Access, DOI: 10.1109/ACCESS.2019.2907793

IoT security



Source: Juniper Research

Main aim of information security

- You spend X so that your opponent has to spend Y to do something you don't want them to do.
  - Y is really greater than X ... and there are many opponents.
- It's a resources game of Time, Money, and Computational Power (== time × $$$)
- Implications:
  - Given enough resources, someone will get in.
  - Given enough attackers, someone will get in.
  - Given enough time, someone will get in.
  - Thus, all systems can, and eventually will, fail.
- The aim is to raise the bar to an adequate level of (in)security for the asset you are trying to protect

Basic principles of security

- **Authenticity**
  - Proof of a message's origin
  - Integrity plus freshness (i.e., message is not a replay)
- **Confidentiality**
  - The ability to keep messages secret (for time *t*)
- **Integrity**
  - Messages should not be able to be modified in transit
  - Attackers should not be able to substitute fakes
- **Non-repudiation**
  - Cannot deny that a message was sent (related to authenticity)
- **Availability**
  - Guarantee of quality of service (fault tolerance)
- **Covertness**
  - Message existence secrecy (related to anonymity)

Other definitions

- **Secrecy**
  - A technical term which refers to the effect of actions that limit access to information
- **Privacy**
  - The ability and/or right to protect the personal secrets of you or your family; including your personal space
- **Anonymity**
  - The ability/desire to keep message source/destination confidential

Trust

- A **trusted** system is one whose failure can break security policy.
- Who can you trust???
  - Software vendors may not fix all the bugs in their software
  - Internet protocols are not secure by design
  - Managers may not invest sufficient resources for security
  - Employees and users may not care about security, choose poor passwords, use wrong configurations, vulnerable to social engineering threats (e.g., phishing)
  - Cryptographic key exchange algorithms may be vulnerable to quantum computers that can factor large numbers

Kerckhoffs's Principle

- Security through obscurity does not work
- Full disclosure of the mechanisms of security algorithms and systems (except secret key material) is the only policy that works.
- Kerckhoff's Principle: For a system to be truly secure, all secrecy must reside in the key.
- If the algorithms are known but cannot be broken, the system is a good system.
- If an algorithm is secret and no-one has looked at it, nothing can be said for its security

Different systems have different security requirements

- Bank security requirements
  - Bookkeeping system: Highest level of integrity
  - ATMs: Physical security, customer authentication
  - Internet Banking: Authentication, availability

- Military security requirements
  - Military communications: Covertness, availability, confidentiality
  - Compartmentalisation: Confidentiality, availability, resistance to traffic analysis.

- Hospital security requirements
  - Online database: Integrity, Privacy
  - Telehealth: Authentication, confidentiality, availability

Securing systems

- A system can be
  - **A product or component**, e.g. software program, cryptographic protocol, smart card
  - **… plus infrastructure**, e.g., PC, operating system, communications
  - **… plus applications**, e.g., web server, payroll system
  - **… plus IT staff**
  - **… plus users and management**
  - **… plus customers and external users**
  - **… plus partners, vendors**
  - **… plus the law, the media, competitors, politicians, regulators…**

- **It's a lot easier to break a system than to make it secure**
  - A demonstration of a failure is sufficient claim the system is insecure
  - Resistance to a host of attacks does not prove the system is secure

Passive and active attacks

- Passive attacks do not involve the modification or fabrication of data, e.g.,
  - eavesdropping on communications,
  - Message interception and release of information,
  - traffic analysis

- Active attacks:
  - Fabrication: An unauthorised party inserts counterfeit objects into the system. Examples include masquerading as an entity to gain access to the system
  - Interruption: An asset of the system is destroyed or becomes unavailable or unusable. Examples include denial-of-service attacks on networks
  - Modification: An unauthorised party not only gains access to but tampers with an asset. Examples include changing values in a data file or a virus

Assessing risks

- Systems often fail because designers:
  - Protect the wrong things
  - Protect the right things in the wrong way
  - Make poor assumptions about their systems
  - Do not understand the threat model properly
  - Fail to account for paradigm shifts (e.g., the Internet)
  - Fail to understand the scope of their system

	Extreme	High	Medium	Low	Negligible
Certain	1	1	2	3	4
Likely	1	2	3	4	5
Moderate	2	3	4	5	6
Unlikely	3	4	5	6	7
Rare	4	5	6	7	7

Risk Assessment Matrix

Summary

- Information security is a resource game
- All systems are buggy, the bigger the system the buggier it is
- Nothing works in isolation
- Know your system, Know your threat model
- Nothing is perfectly secure, All systems can, and eventually will, fail
- Humans are most often the weakest link
- It's a lot easier to break a system than to make it secure

Hash Functions and MACs

THE UNIVERSITY OF SYDNEY

Cryptography

- **Cryptography** is the study of mathematical techniques related to the design of ciphers.
- Crypto building blocks are otherwise known as cryptographic primitives. For example:
  - hash functions
  - block ciphers
  - stream ciphers
  - digital signatures
  - random number generators

Basic Concepts

- A function $f: f: X \to Y$ is defined by:
  - The **domain**, a set $X = \{x_1, x_2, \ldots, x_n\}$.
  - The **codomain**, a set $Y = \{y_1, y_2, \ldots, y_m\}$.

- When $f: X \to Y$ is a function:
  - The **image** of $x \in X$ is called $f(x)$, an element of Y.
  - The **range** of f is the set of all images and is a subset of Y.
  - If $f(x) = y$, then $x = f^{-1}(y)$ is called a **preimage** of y.

- Example: let $f: \{-1, 0, 1\} \to \{0, 1, 2\}$ be defined by $f(x) = x^2$.
  - $f(-1) = 1, f(0) = 0$, and $f(1) = 1$.
  - The preimage of 1 is $f^{-1}(\{1\}) = \{-1, 1\}$.
  - The preimage of 2 is $f^{-1}(\{2\}) = \{\}$.
  - The range of f is $\{0, 1\}$.

One Way Functions

- Consider the set of all binary strings of length n, i.e. $\{0,1\}^n$. For example:
  - $\{0,1\}^1 = \{0,1\}$.
  - $\{0,1\}^2 = \{00,01,10,11\}$.
- A function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ is a one-way function (OWF) if:
  - It is "easy" to compute $f(x)$ for all $x \in X$
  - It is "computationally infeasible" to find a preimage.
- **Intuitively:**
  - Given x, it is easy to compute $f(x)$
  - Given $f(x)$, it is hard to compute x
- Smashed Plate Example
  - Write a message on the side of a plate: x; Smash the plate: $f(x)$; Finding the inverse $f^{-1}(x)$ is difficult (but not impossible)
- Data Encryption Standard Cipher
  - $f(x) = \mathrm{DES}(m,k) = c$; Given c, it is difficult to find k and m.

Hash Functions

THE UNIVERSITY OF SYDNEY

Hash Functions

- A hash function, h, is an efficiently computable mapping of arbitrarily long strings to short fixed length strings.
- **Minimum properties:**
  - Compression
    - Typically, any number of bits to $\leqslant 512$ bits
    - e.g., MD5, SHA256, SHA512
  - Ease of computation
    - Given h and x, $h(x)$ is usually easy to compute.
- **Keyed Hash Functions**
  - Some hash functions take both a key (k) and a message (m)
  - $\mathrm{MAC}_k(m) = h(m, k)$
  - They are also called message authentication codes (MAC) or hash-based message authentication codes (HMAC).

Properties of Secure Hash Functions

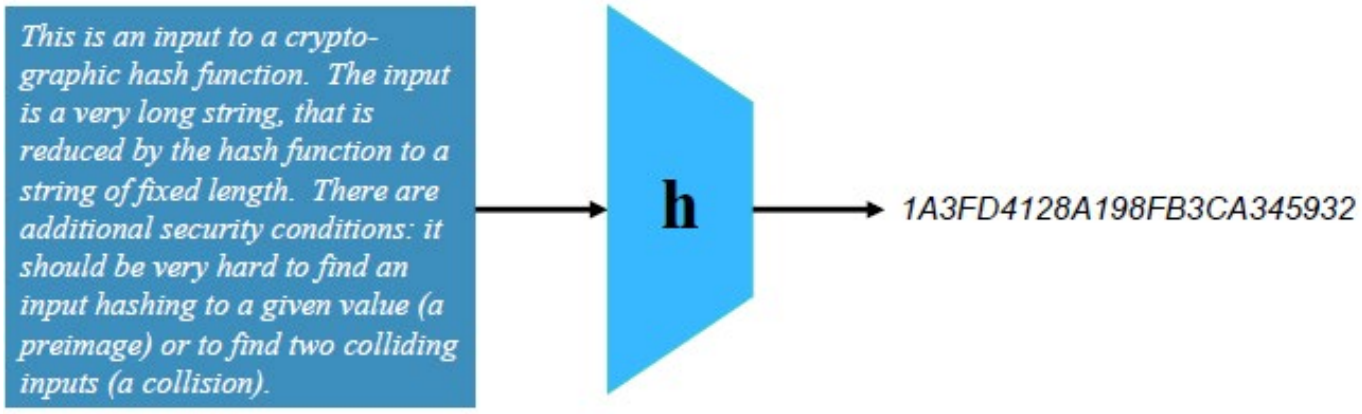
- Let $h: X \rightarrow Y$ be a hash function. In order to be secure, it must satisfy the following properties:
  - #1 Preimage Resistance
    - Given y it is "hard" to find a preimage x such that $h(x) = y$.
  - #2 Second Preimage Resistance
    - Given a particular x (and hence y), it is "hard" to find $x' \neq x$ such that: $h(x') = h(x) = y$.
  - #3 Collision Resistance
    - It is "hard" to find any pair $x \neq x'$ such that $h(x) = h(x')$.
    - Note: #3 ⇒ #2 since !#2 ⇒!#3
- A one-way hash function satisfies #1 and #2
- A collision resistant hash function satisfies #3 (and hence #2)

Properties of Secure Hash Functions

- Hash functions are extremely useful for confirmation of knowledge without revealing what you know.
  - Rather than sending Alice a secret across the Internet, just send the hash of it. If Alice knows the secret, she can hash it and verify that you know it too. This is much safer and efficient than sending the secret, which can be intercepted (provided the hash function is strong).
  - Example 1: Password Files
    - Store the hash of a password. When someone tries to log in, just compute the hash and compare to what was stored. If the password file gets stolen, the hash needs to be reversed before the attacker can use the passwords ("cracking passwords").
  - Example 2: Virus Protection & Host Intrusion Detection
    - For each file x, $h(x)$ is stored off system. Periodically hash all files and check the hashes match. Property #2 is critical as it should be hard to find x' such that $h(x) = h(x')$ (otherwise viruses could hide)

Real World Hash Functions

Name	bits	h("ELEC5616")
MD5	128	df5e182130aa5448f78d79103e4203e3
SHA-1	160	e7e880966d3772f00c3dc6ba12bb68660b003ee0
SHA256	256	943b8fa48772fbb052a225fccbe1adabe0bbbd05eb162c536c5da0213f84971b
SHA512	512	449af7edb504b62dc3422a283b3e4fa5f3f22a0e56ca60bd8e57cdf4023df72965243463d074657f8a6355e98ef8d9d9170f00dcb4198ae22720121356509bc9



Pre-Image Resistance Attack

- To **brute force** in cryptanalysis is to search the entire space of possible alternatives. A subset of this is a **dictionary attack** where we throw subsets of the keyspace (dictionaries) at the problem.

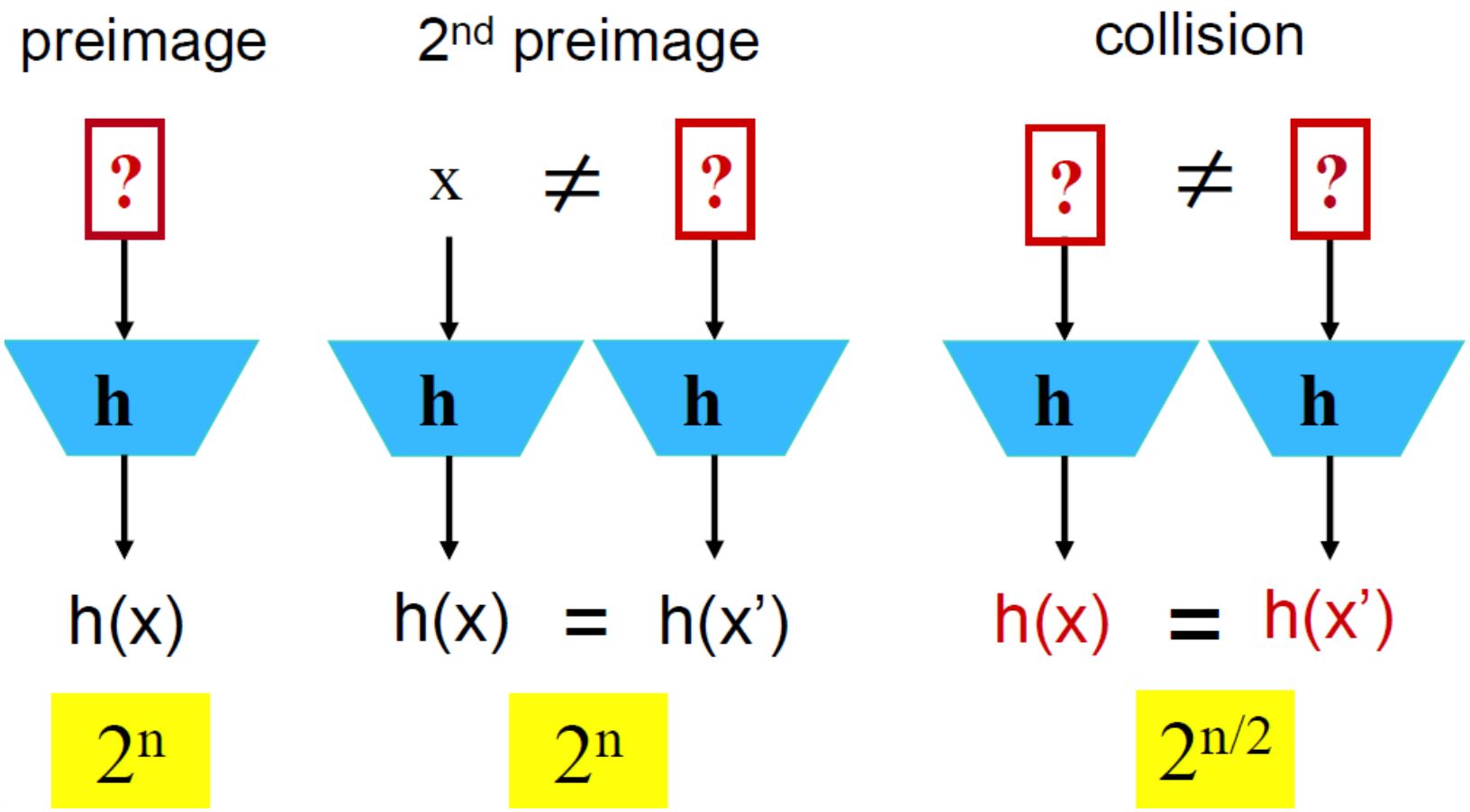
- We can use brute force to attack **pre-image resistance**:
  - Say a hash produces an n-bit output: $y = h(x)$
  - We must try 2^{n-1} hashes before $\Pr\{h(a) = y\} \geqslant 0.5\ (a \in Z)$
  - Intuitively: if the secret key is in one of ($2^{10} = 1024$) boxes, you have to open half of them ($2^9 = 512$) on average before you find the secret key.

Collision Resistance Attacks

- A **birthday attack** is an attack on collision resistance:
  - How many people must be in a room such that a pair of 2 share a birthday?
    - Pr[two people same birthday] > 0.5?
    - Requires only 23 people
    - https://en.wikipedia.org/wiki/Birthday_problem
- For an n-bit hash, we must try $\mathbf{2^{n/2}}$ **hashes** of random messages on average before the collision attack succeeds.
  - SHAttered: SHA-1 Collision Attack
    - https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

- Hash function security summary – Wikipedia
- Hash Functions | CSRC (nist.gov)

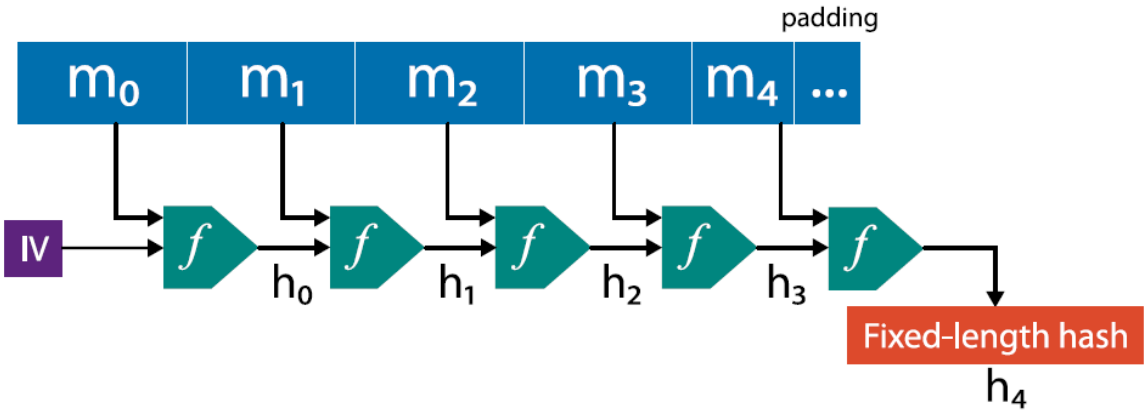
Security Requirements of n-bit Collision Resistance Hash Functions (CRHF)



https://homes.esat.kuleuven.be/~preneel/preneel_hash_ecrypt_june2011.pdf

Iterated Hash Construction

- Merkle–Damgård construction is used to build MD5, SHA-1, and SHA-2.



  - IV is initialization vector and f is a one-way compression function.
  - Simply divide a message M into n r-bit blocks.
  - Length must be a multiple of a fixed number (e.g 512 bits), so the message must first pass through a padding function.
- Sponge construction is used for SHA-3
  - https://keccak.team/keccak.html

Example outputs

MD5

Input	Hash Value (as hex byte string)
""	d41d8cd98f00b204e9800998ecf8427e
"a"	0cc175b9c0f1b6a831c399e269772661
"abc"	900150983cd24fb0d6963f7d28e17f72

SHA-1

Input	Hash Value (as hex byte string)
""	da39a3ee5e6b4b0d3255bfef95601890afd80709
"a"	86f7e437faa5a7fce15d1ddcb9eaeaea377667b8
"abc"	a9993e364706816aba3e25717850c26c9cd0d89d

Message Authentication Codes (MACs)

THE UNIVERSITY OF SYDNEY

Keyed Hash Functions (MACs)

- Message Authentication Codes (MACs) use a one-way hash function with the addition of a key:
  - $h_k : \{0,1\}^* \rightarrow \{0,1\}^n$ The key is secret and necessary to verify the hash $h_k(m)$ and can be thought of as a cryptographic checksum.

- Goal:
  - Provides message authentication where sender and receiver share a secret key.
  - An eavesdropper cannot fake a message with a valid MAC.
  - Used for message integrity, not message secrecy.

Properties of MACs

- Given m and k it is easy to construct $h_k(m)$.
- Given pairs of messages and MACs $\left(m_x, h_k(m_x)\right)$ it is hard to construct a valid new pair $\left(m_y, h_k(m_y)\right)$ for $m_x \neq m_y$ without the knowledge of k.

- Example 1
  - Alice and Bob share a secret key k
  - An adversary can't send a message with a valid MAC
  - $\mathrm{MAC}(m) = h_k(m)$

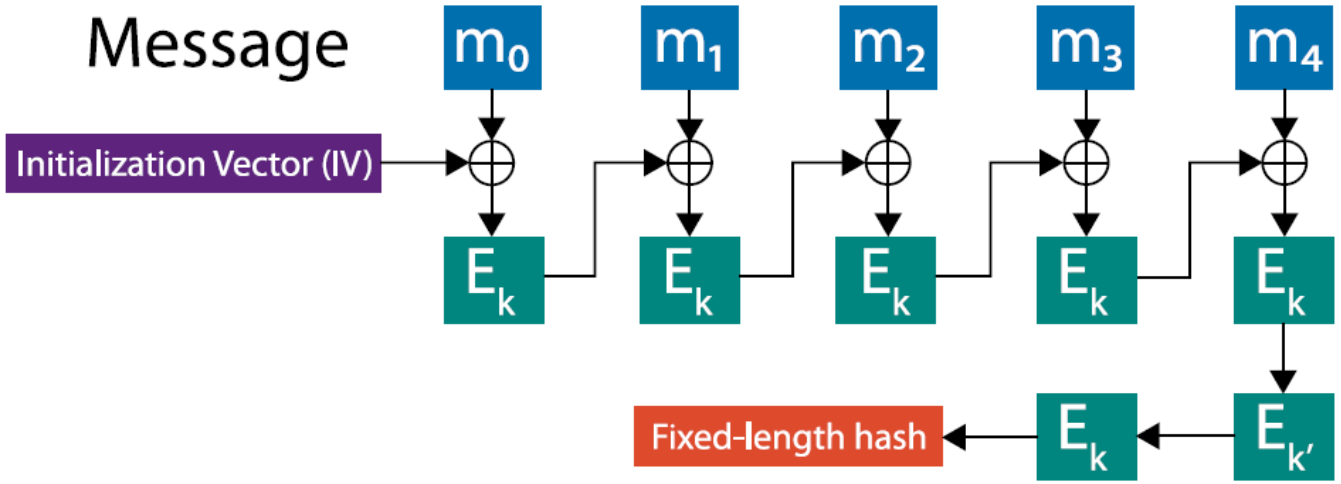
- Example 2
  - Say a hash function is used for virus protection and stores the signatures for each file in a database. *Couldn't the virus also modify the database?* **With a MAC, the virus can't because it doesn't know the key.** If it had write permissions, it could however corrupt the database or replace the verification program with a trojan/fake.

Examples: Hash Based MAC (HMAC)

- MAC based on non-keyed hash function h
  - Attempt 1: $\mathrm{MAC}_k(m) = h(k \mid m)$
    - Insecure: attacker can arbitrarily add to the end of the message. (Merkle-Damgård construction)
  - Attempt 2: $\mathrm{MAC}_k(m) = h(m \mid k)$
    - Insecure: vulnerable to the birthday attack!
  - Attempt 3: $\mathrm{MAC}_{k,k'}(m) = h(k \mid m \mid k')$
    - More secure: enveloped method
  - BEST: $\mathrm{MAC}_k(m) = h\Big((k \oplus \mathrm{opad}) \mid h\big((k \oplus \mathrm{ipad}) \mid m\big)\Big)$
    - opad is the outer padding (0x5c5c5c5… one block of hex constant)
    - ipad is the inner padding (0x3636363… one block of hex constant)
    - RFC2104 https://www.ietf.org/rfc/rfc2104.txt

Examples: Cipher-Based MAC (CBC-MAC)

- CBC-MAC uses a technique known as Cipher Block Chaining (CBC).
  - Turn a message into blocks
  - Repeated encryption using a block cipher is XORed ($\oplus$)
  - Secret key $= (k, k', IV)$, IV: Initialisation Vector (random bits)



  - If E is a MAC then CBC-E is also a MAC. Often used in the banking industry.

Questions

- What is a secure hash function?
- What are secure hash functions used for?
- What is a brute force attack? How can we prevent it?

THE UNIVERSITY OF SYDNEY