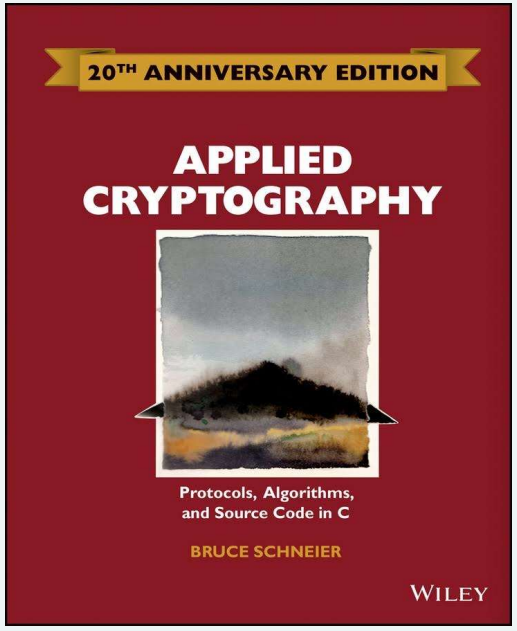
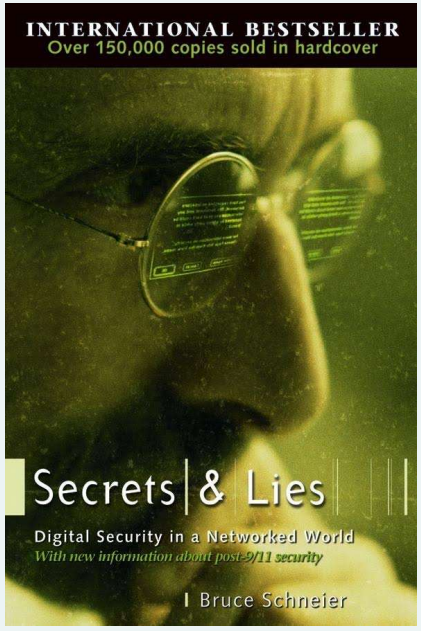
Overview

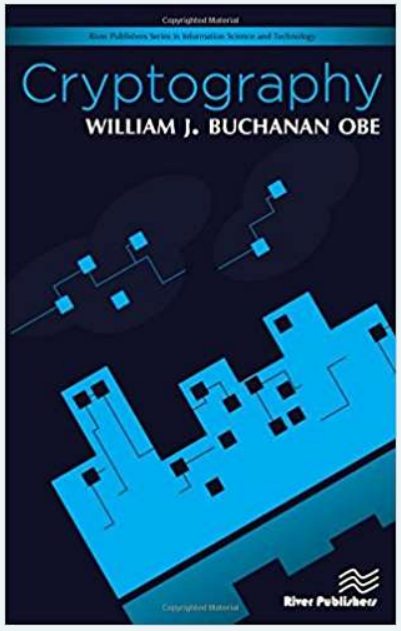
- Understand the concept of security in large systems
- Security fundamentals
- Two models of information security
- CIA of security

Books

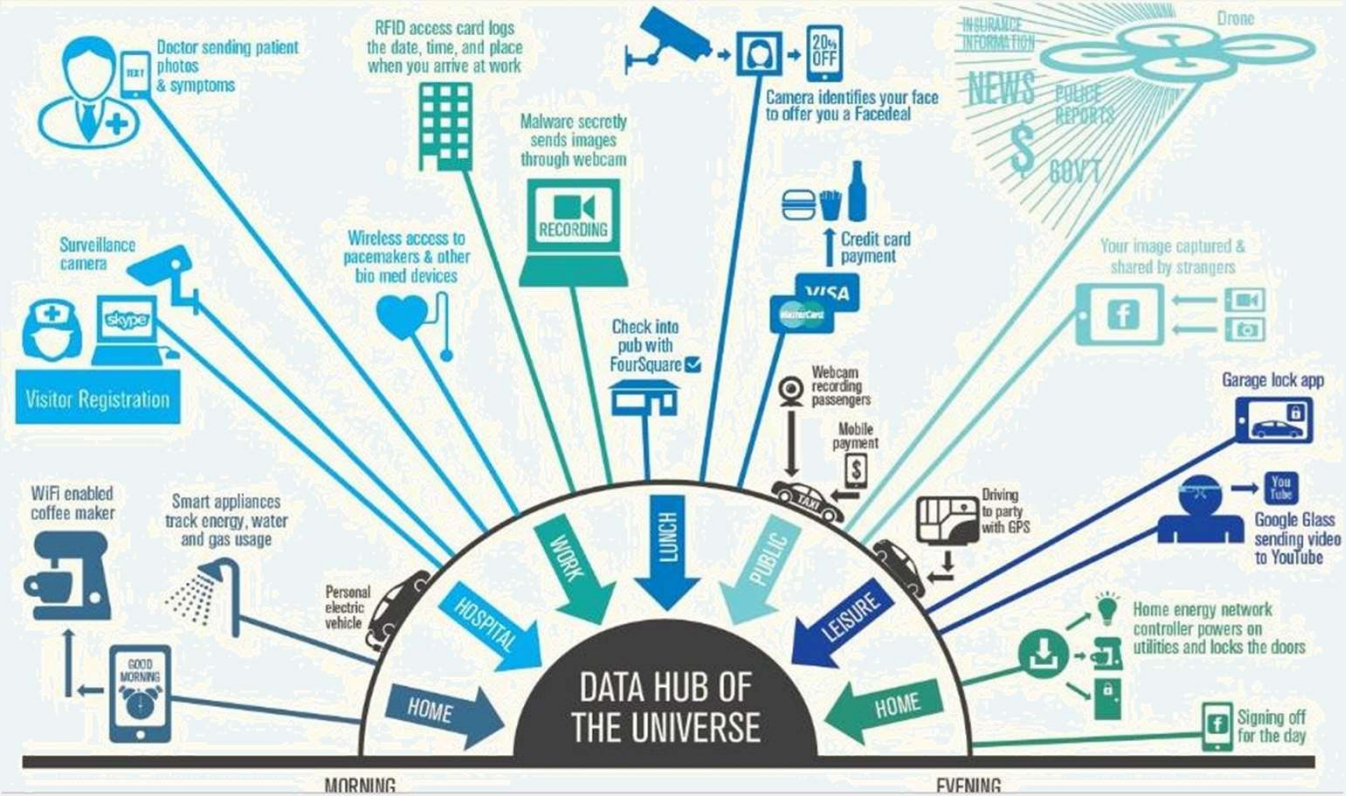
- I like these books, you might too
  - interest only







We live in an integrated world



Source: http://privacyguidance.com/

Network Security

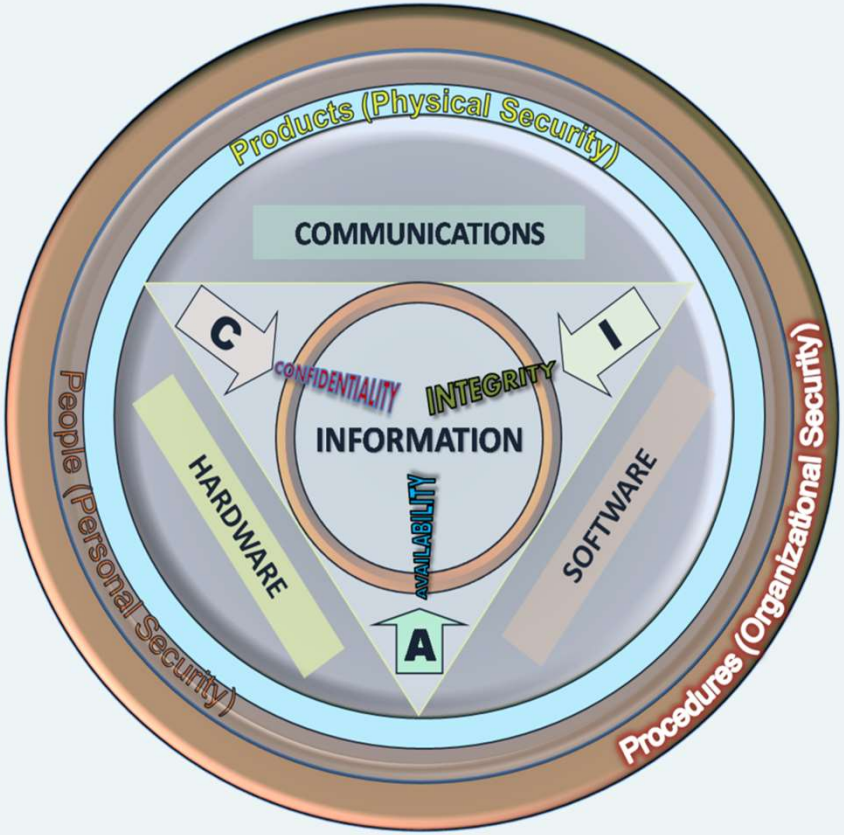
- Network security is about:
  - Security and privacy
    - Confidentiality
    - Integrity
    - Availability
    - Non-repudiation

- Through
  - Physical security
  - Virtual security
  - Data security
- Via
  - Detection
  - Prevention
  - Correction

Information Security

- CIA Model
  - Confidentiality
  - Integrity
  - Availability
- CAIN Model
  - Confidentiality
  - Integrity
  - Availability
  - Nonrepudiation

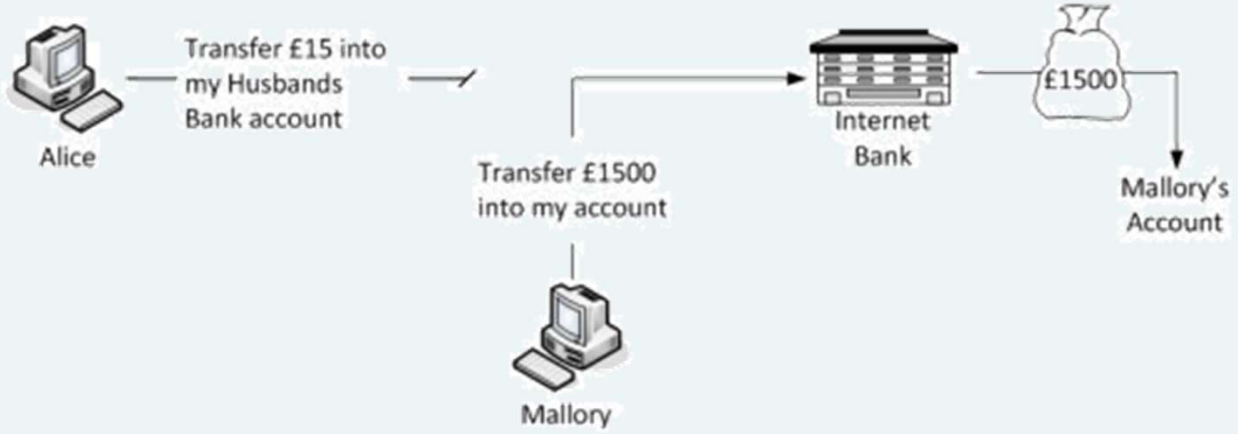
Three ring structure



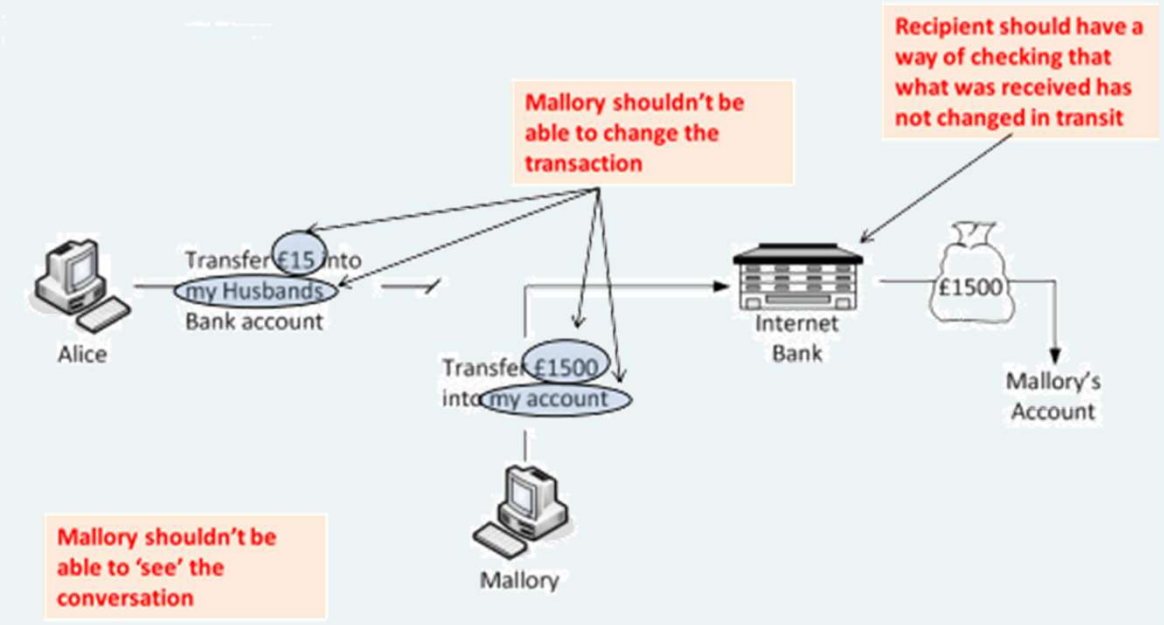
Key Concepts

- Confidentiality
  - (covers both data confidentiality and privacy): preserving authorised restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information. A loss of confidentiality is the unauthorized disclosure of information.
- Integrity
  - (covers both data and system integrity): Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity. A loss of integrity is the unauthorized modification or destruction of information.
- Availability
  - Ensuring timely and reliable access to and use of information. A loss of availability is the disruption of access to or use of information or an information system.
- Additional concepts
  - Authenticity
    - The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, a message, or message originator.
  - Accountability
    - The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity.
  - Auditing
    - Ensuring there is a trail of activity such that actions and owners are tracked and cannot repudiate their actions

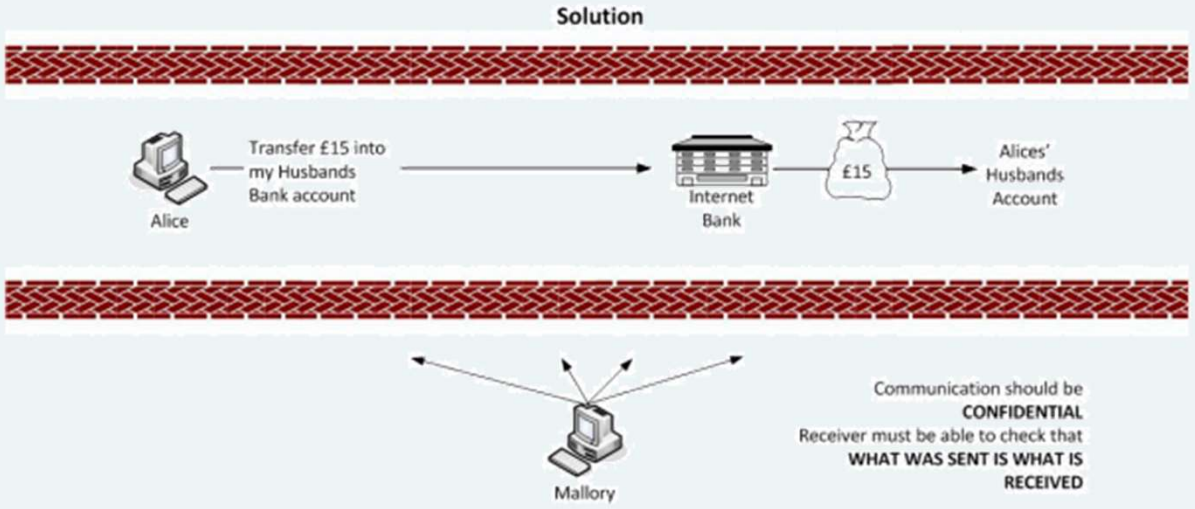
Simple Example



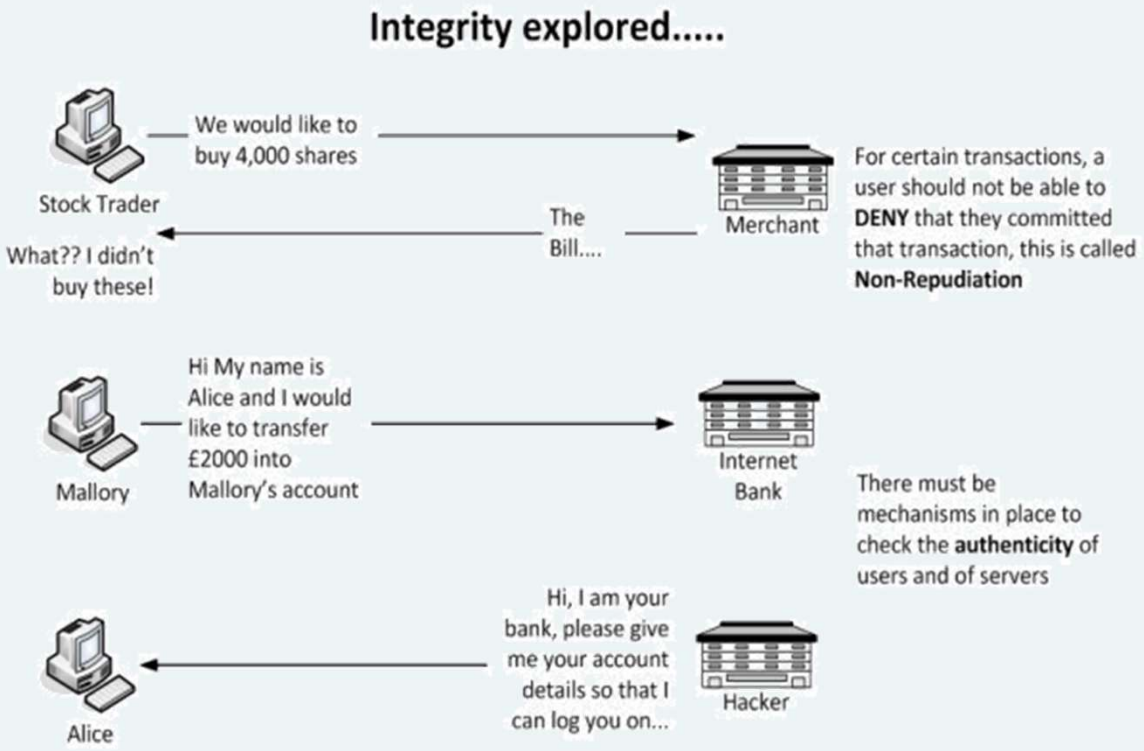
CIA: Confidentiality



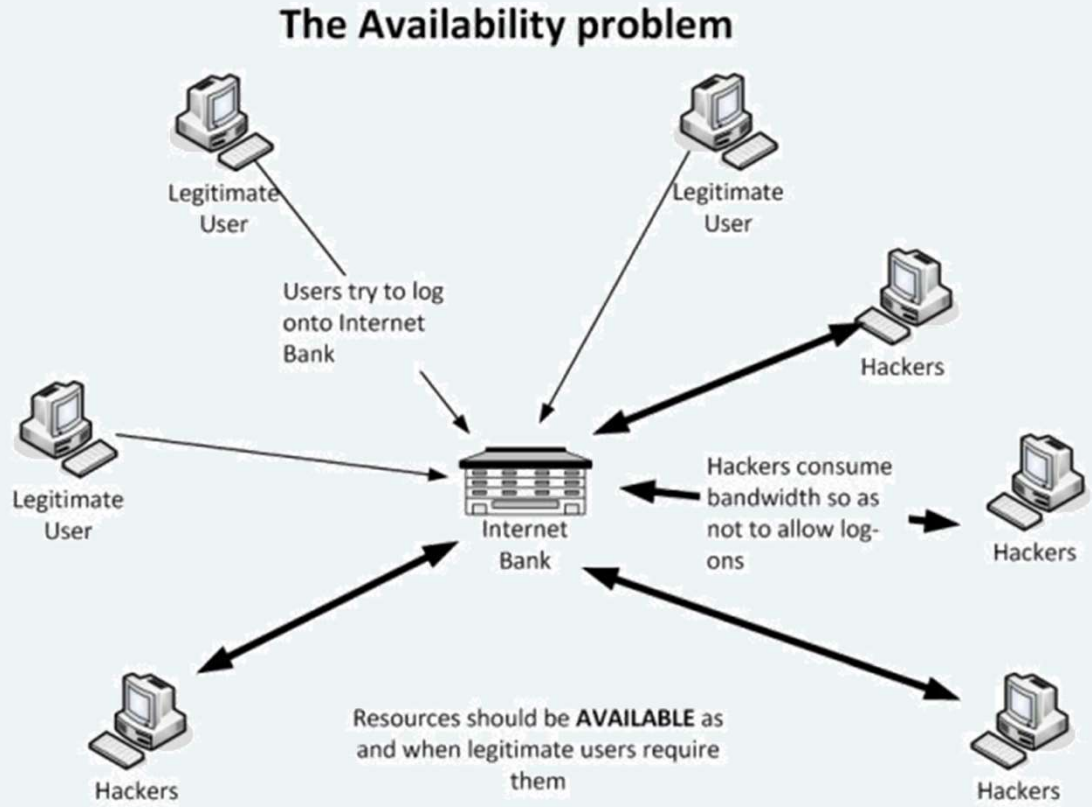
CIA: Confidentiality



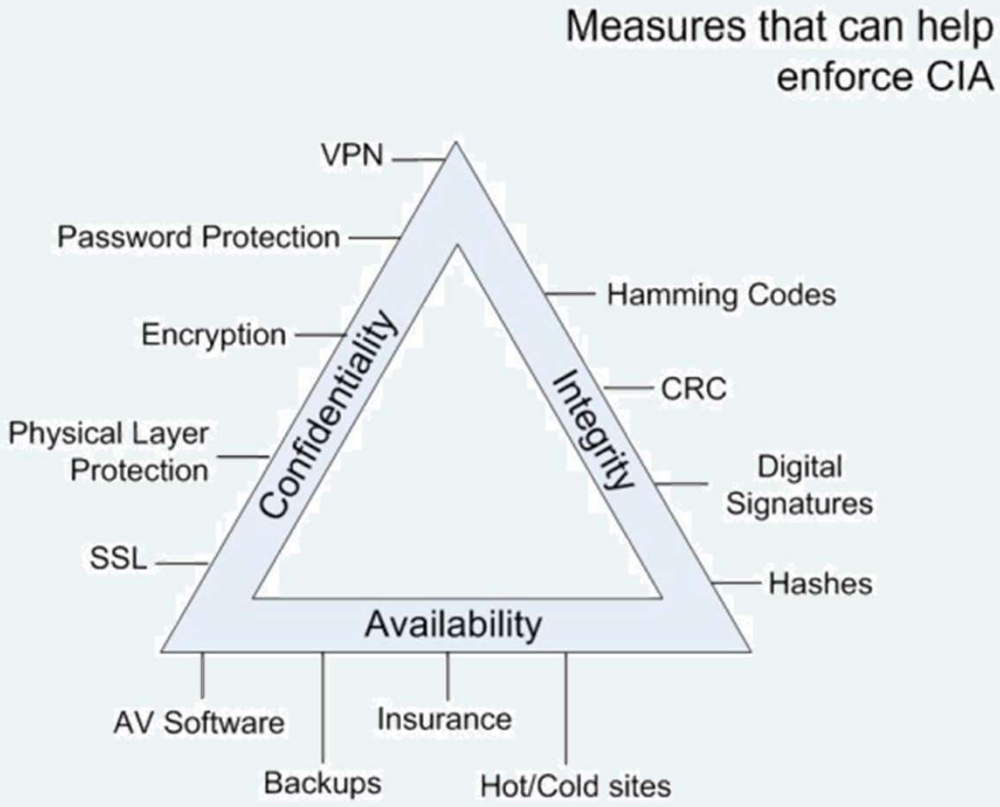
CIA: Integrity



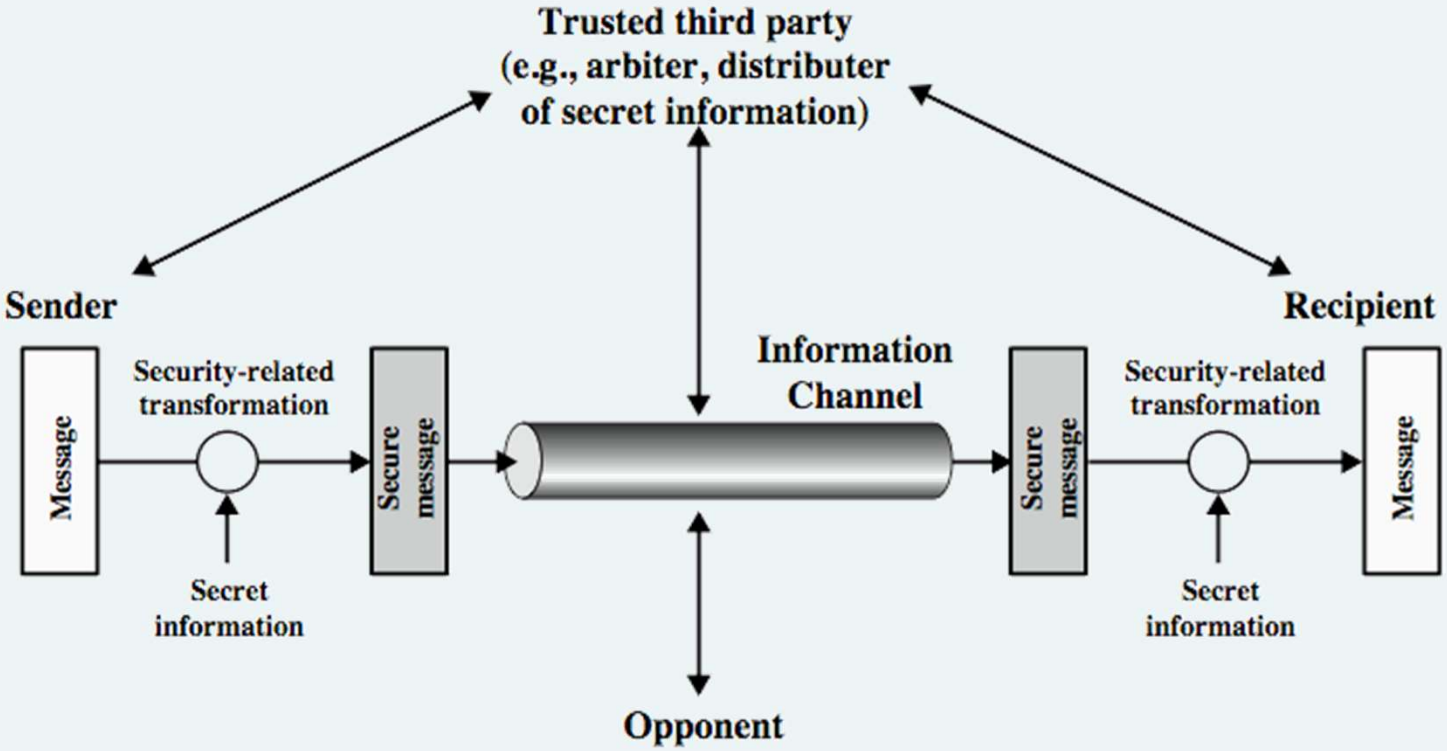
CIA: Availability



Enforcing CIA



Information Security

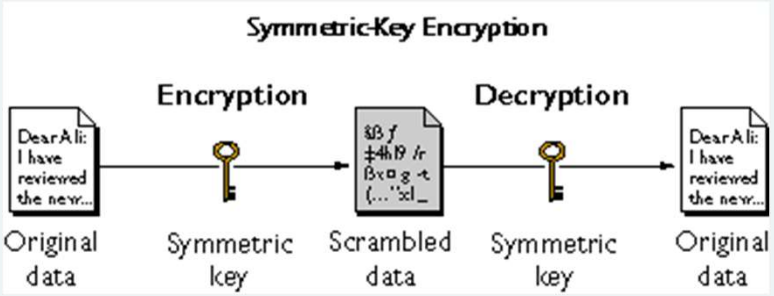


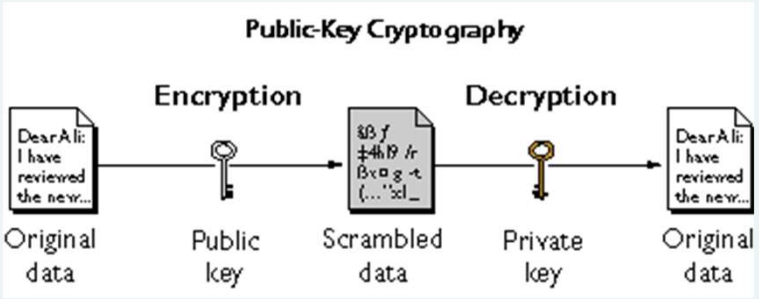
Initial concepts and terms for Cryptography

- **Plaintext** – the original data (un-encrypted)
- **Ciphertext** – encrypted data
- e.g. kmM9VB4wH/T8I4AgIRqB/XppL7YlL2EnqhdYMCh
- **Encryption** – the process of converting plaintext into ciphertext
- **Decryption** – the process of reversing the encryption process in order to reproduce the original data
- **Cryptosystem** – an algorithm, combined with the key which performs the encryption
- **Key** – a binary number which is used to perform the encryption
  - The larger the number, the better the encryption
  - The size is known as the key space
- **Cryptanalysis** – the process of attempting to calculate/guess the key

Encryption

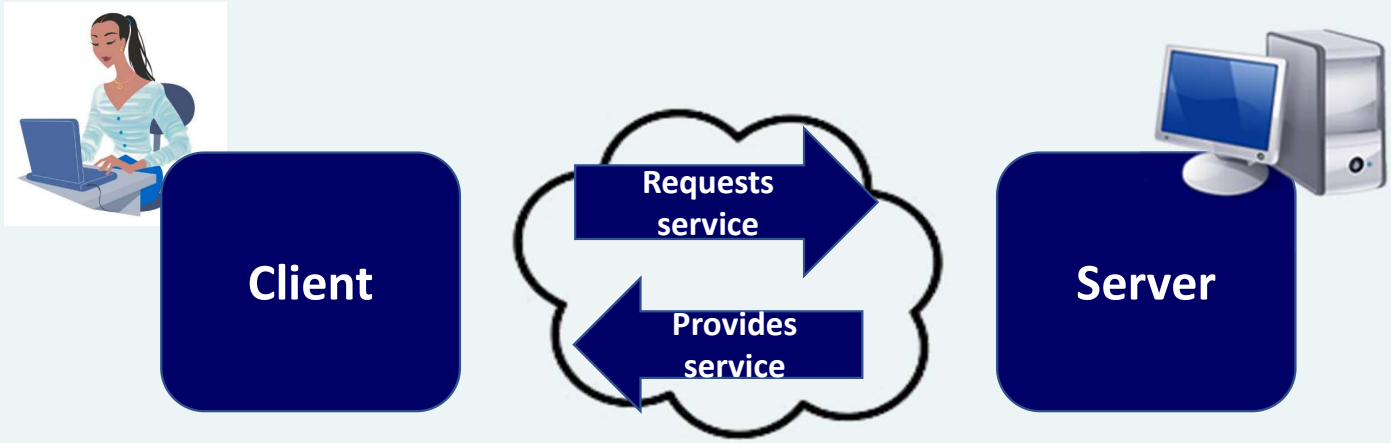
- Encryption algorithms use logical functions to perform the encryption
- There are a number of commonly used encryption models:
  - Secret shared keys/Symmetric Encryption
  - Public Key/Asymmetric Encryption
  - Hashing/one way functions





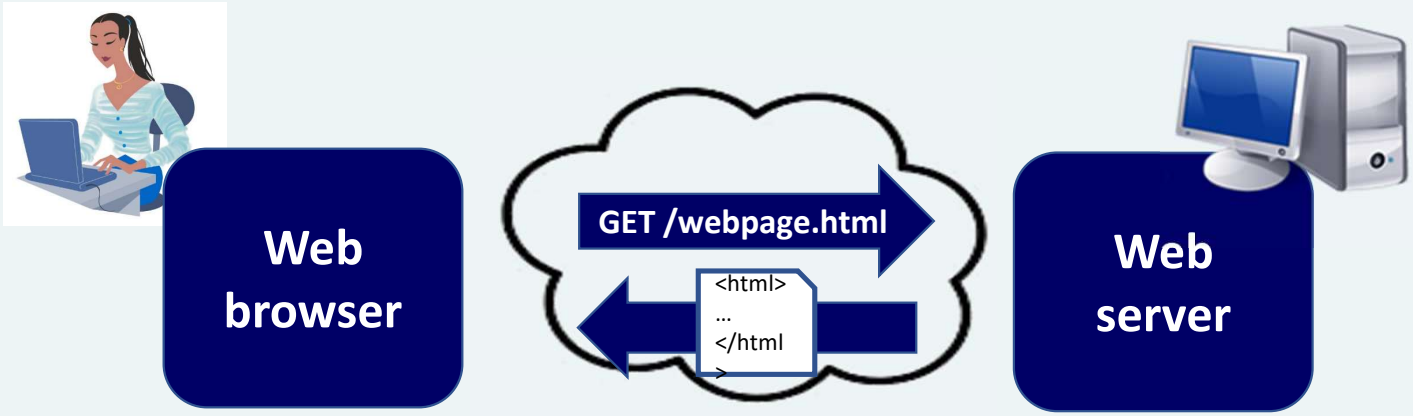
Client-server communication

Your Assessment?



Client-server communication – example

- HTTP protocol



Security threat 1: Eavesdropping/snooping

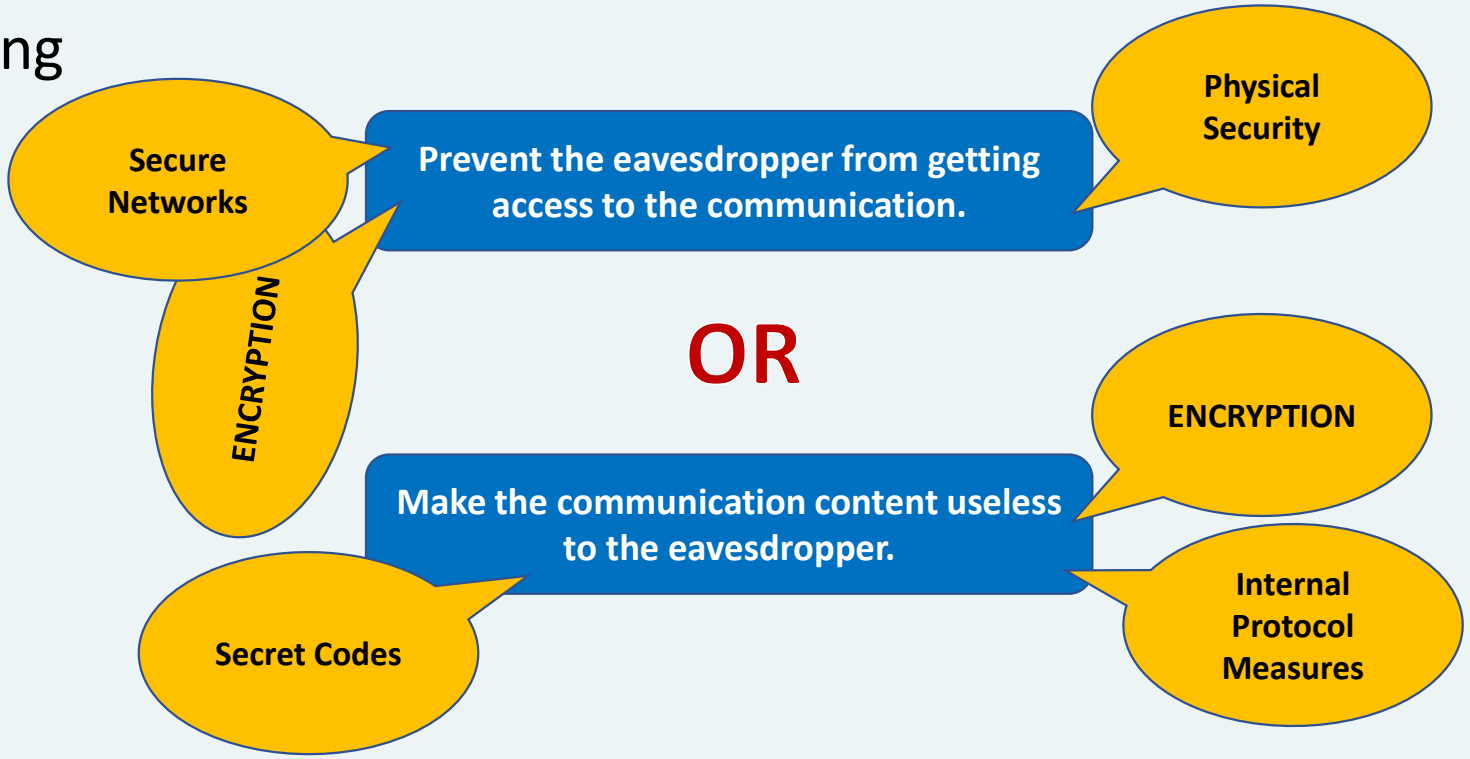
- **Confidentiality problem**

Eavesdropping happens when someone is able to 'listen' the content of a communication.

It is a confidentiality problem: unauthorized access to data.

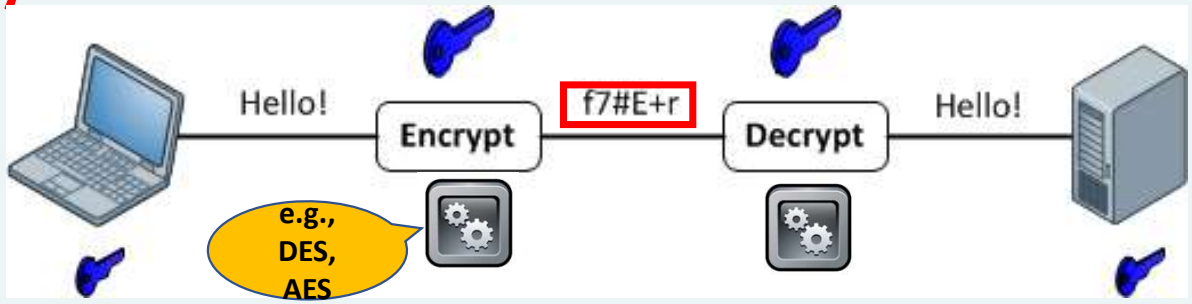
Solutions to eavesdropping

- snooping



Symmetric encryption
(also called secret key encryption)

- **secret key is shared.**



All participants of the communication need to have the secret key to read the plaintext content "Hello".

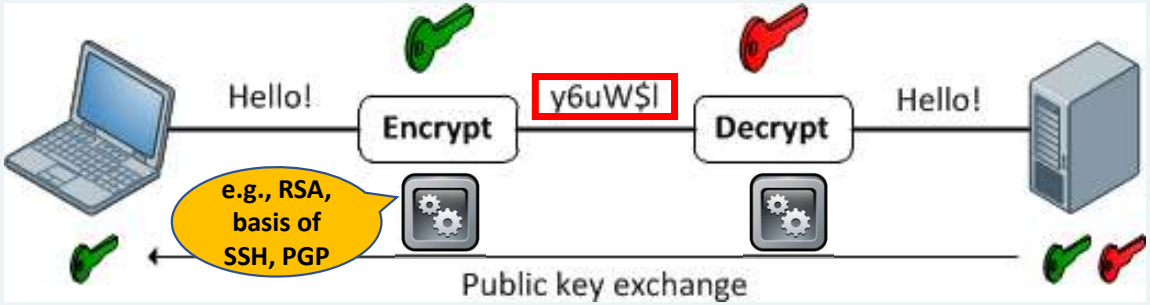
(Source: http://packetlife.net/blog/2010/nov/23/symmetric-asymmetric-encryption-hashing/)

Symmetric encryption: weaknesses

1. All participants on the communication need to have the secret key
   → hosts need it pre-configured
   → it creates several points for compromise
2. The content is only protected as long as the key is kept secret
   → Anyone with access to the secret key can use it to read the communication content
3. It is a challenge to securely share the secret key

Asymmetric encryption (also called public key encryption)

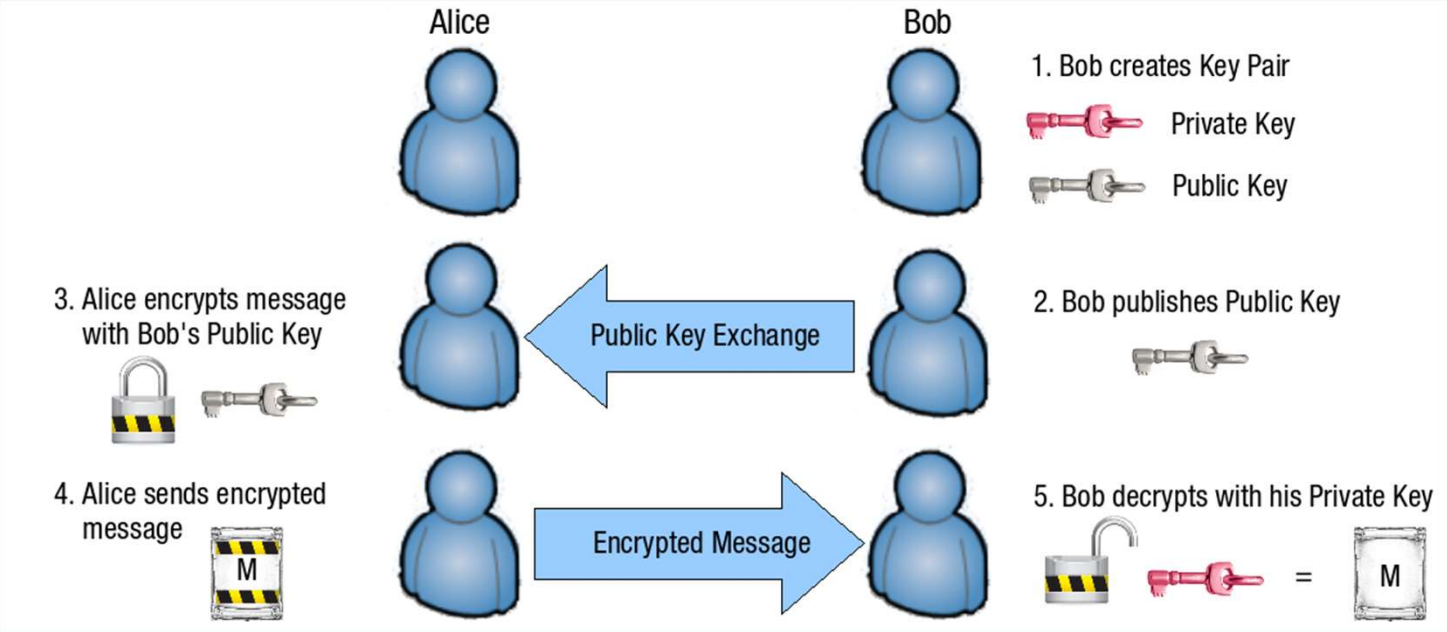
- **Based on two keys: private and public**



Public key is shared and private key is used for decryption to ensure confidentiality.

Asymmetric encryption: to ensure confidentiality

- CIA



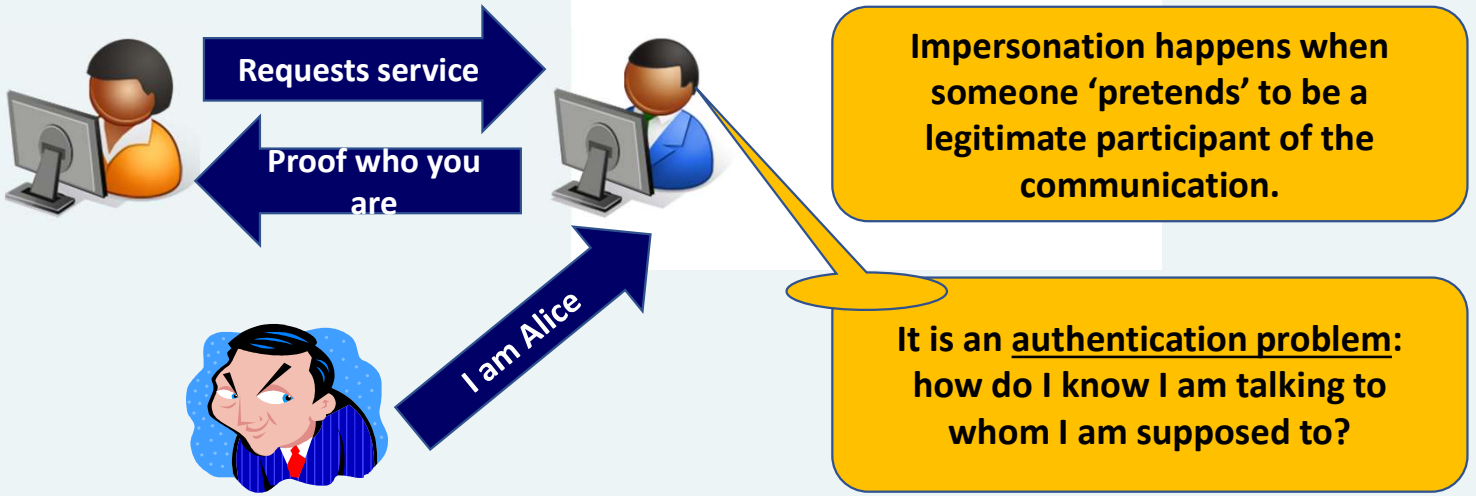
(source: http://documentation.axsguard.net/manuals/Gatekeeper/7.7.0PL2/html/ipsec/)

Conclusion about eavesdropping/snooping

- **Against eavesdropping, we can use symmetric or asymmetric encryption (as described).**
- **Security is as strong as the encryption algorithm and the protection of the secret or private keys.**

Security threat 2: impersonation

- Authentication problem?



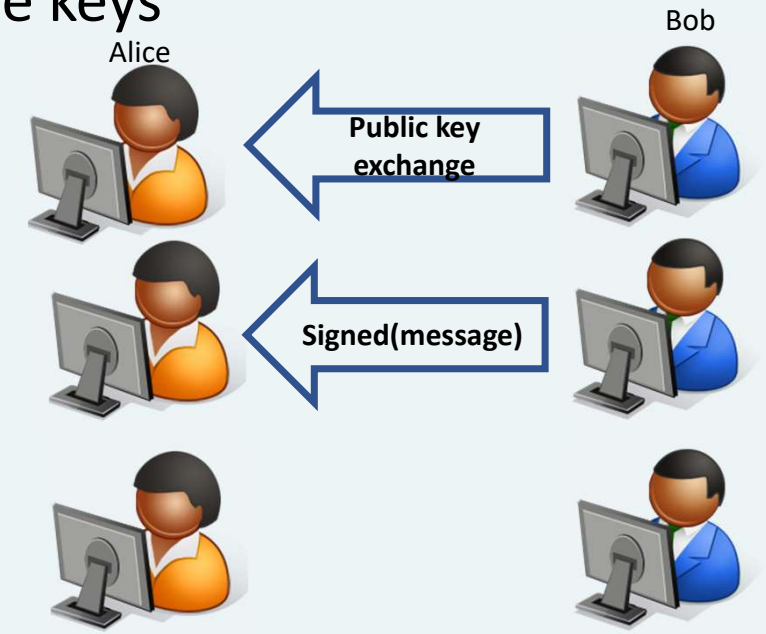
Security threat 2: impersonation

- Who are you?

As a solution she could try something ‘unique’ to her… but still.

Asymmetric encryption used as *digital signature*

- Public and Private keys



1. Bob creates:
   - Public key
   - Private key

2. Bob encrypts message with his private key

3. Alice decrypts the message with Bob's public key

Digital signature: to ensure authentication & non-repudiation

- Digital signature:

  (Bob is the owner of his private key which is kept secret)

  - It ensures authenticity of the message

    → Alice can be sure that the message is coming from Bob

  - It also ensures non-repudiation of the message

    → Bob cannot deny he has sent the message to Alice

- But, on its own, it doesn't ensure confidentiality of the message

  → Since anyone who has Bob's public key can read the message

Conclusion about impersonation

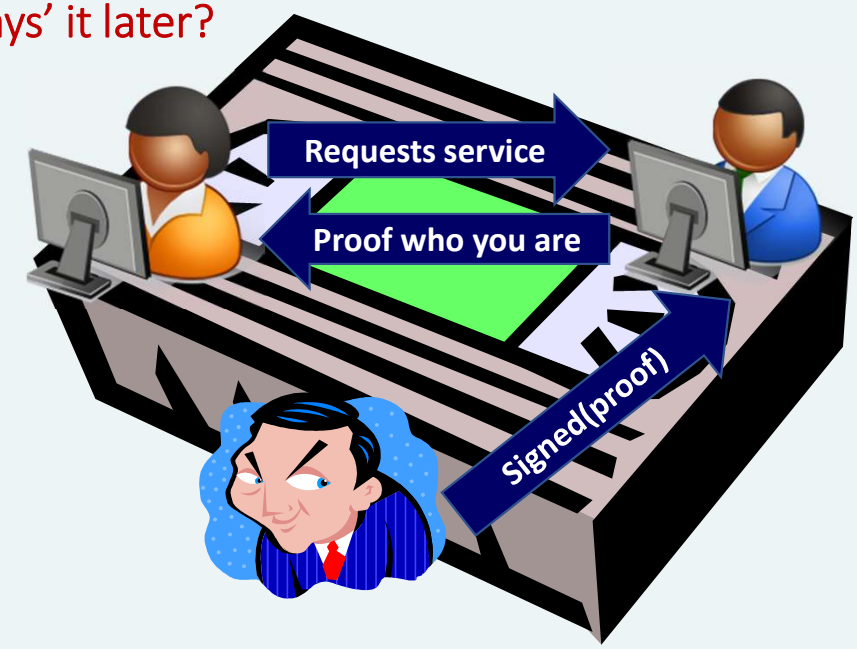
- **Against impersonation in communication we can use digital signature.**

Security threat 3: message replay

What if…. Someone ‘records’ the signed proof.

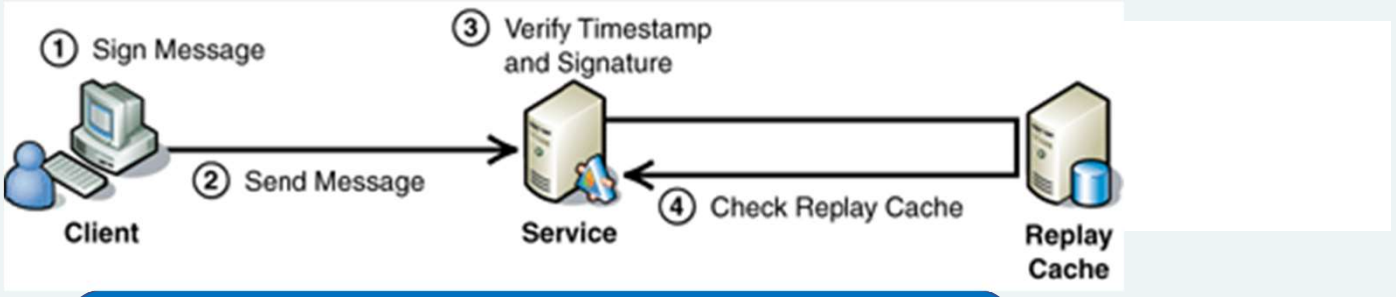
Security threat 3: message replay

…. and ‘replays’ it later?



Message replay detection

- Check Replay Cache



1. Client signs a message with private key
2. Client sends the signed message
3. The service verifies if the signature is valid and compares timestamp with its own clock
   - if signature is invalid OR
   - if it was received beyond acceptable time

spam

→ the message is rejected

Solution to message replay

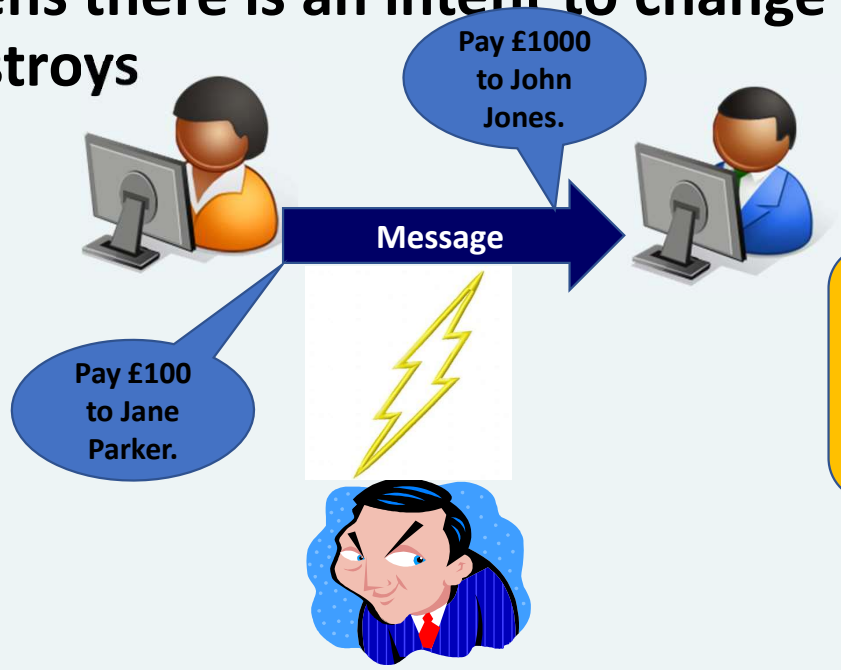
- There is a need to ensure messages are not replayed

Conclusion about message replay

- **Message replay can be detected with digital signature + timestamp.**
- **Other types of sequence numbers could also be used, e.g., session token, one-time passcode.**

Security threat 4: message tampering

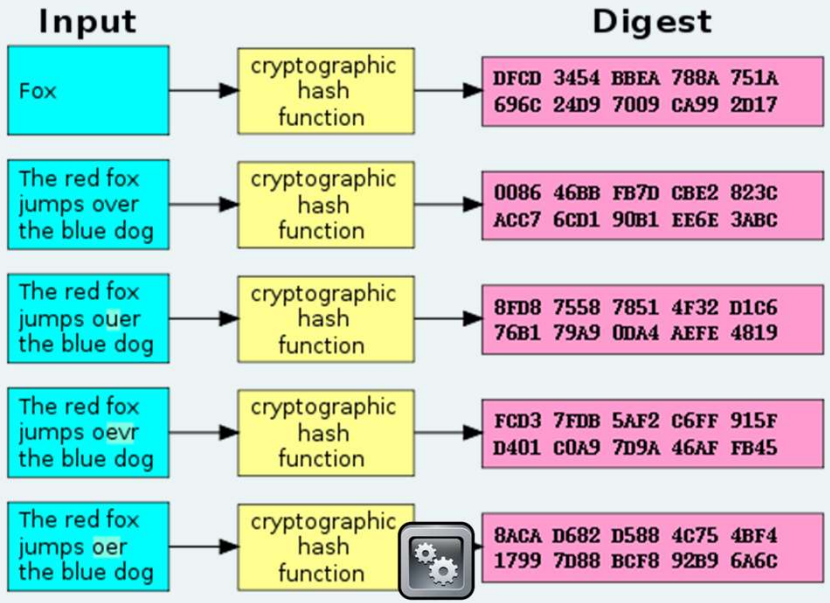
- **happens there is an intent to change a message content in any way or destroys**



Message tampering happens when someone changes its content in any way or destroys/deletes it.
It is an integrity problem.

Solution to message tampering: hashing

Changing a single bit in *x* will result in a different value for *y*



Idea behind hashing

For input data *x,* the output *y* is

$$y=h(x)$$

- Compression
  - Hashing algorithms produce a fixed size output of *y* regardless of the size of *x*. *y* is a fixed length, but needs to be 'small' for this to work
- Efficiency
  - The computation of *h(x)* should be efficient. It will grow for greater lengths of *x*, but shouldn't take much longer
- One way
  - It should be difficult to invert the hash, i.e. Given *y* it should be difficult if not impossible to calculate *x*

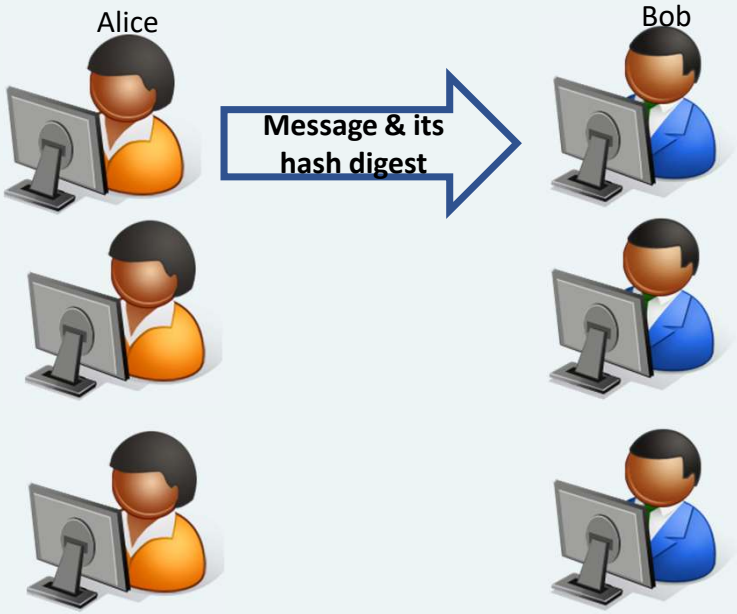
Hashing algorithms

- Example hashing algorithms and digest size
  - MD5 hash algorithm produces 128-bits digests
  - SHA-1 hash algorithm produces 160-bits digests
  - SHA-256 hash algorithm produces 256-bits digests
  - SHA-512 hash algorithm produces 512-bits digests

  ...

Hashing in practice

1. Alice hashes a message with MD5 algorithm



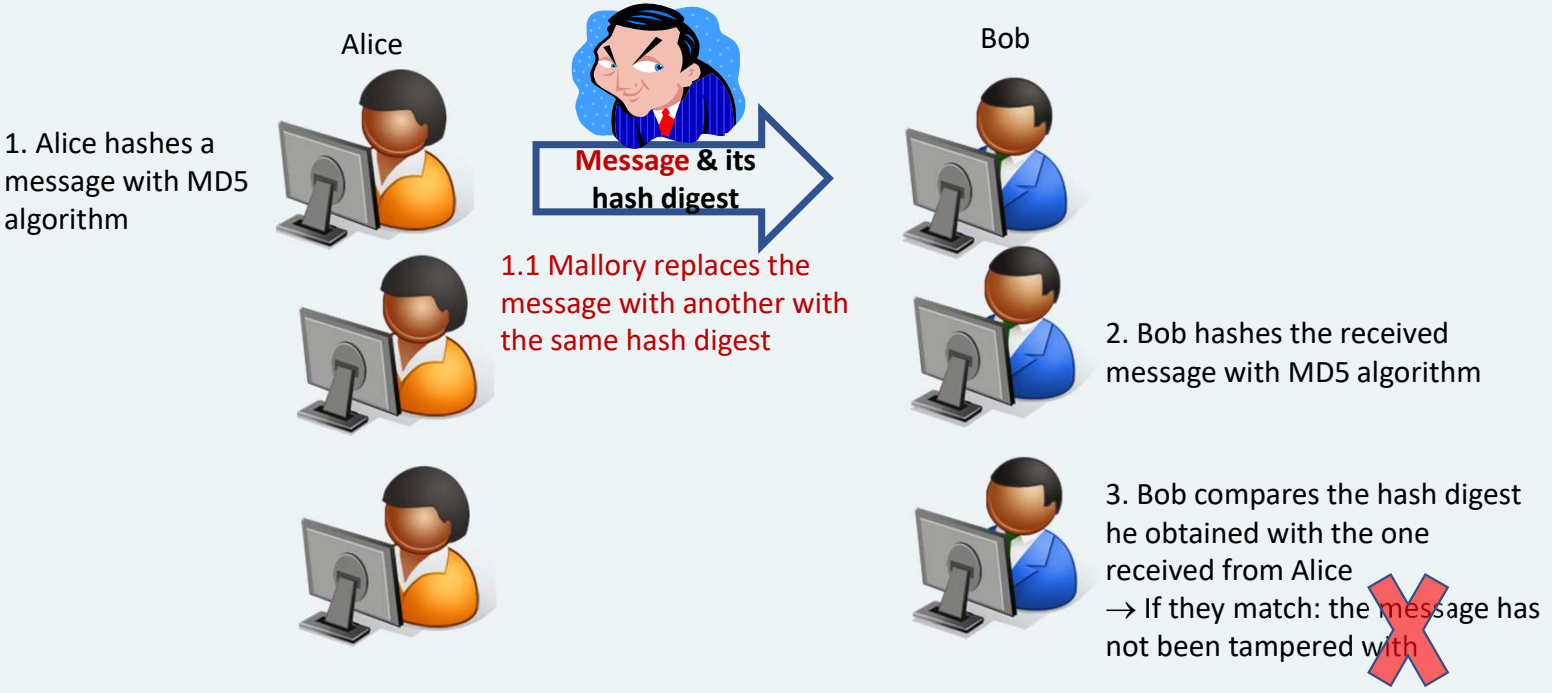
2. Bob hashes the received message with MD5 algorithm

3. Bob compares the hash digest he computed with the one received from Alice
→ If they match: the message has not been tampered with

Potential problem with hashing: collision

- A collision happens when two different messages can be manipulated to generate a same hash digest
- A collision attack
  - on the MD5 hashing algorithm is known since 2004 (Wang et al. 2004)
  - on the SHA-1 hashing algorithm is known since 2005 (Wang et al. 2005)

Idea behind collision attack



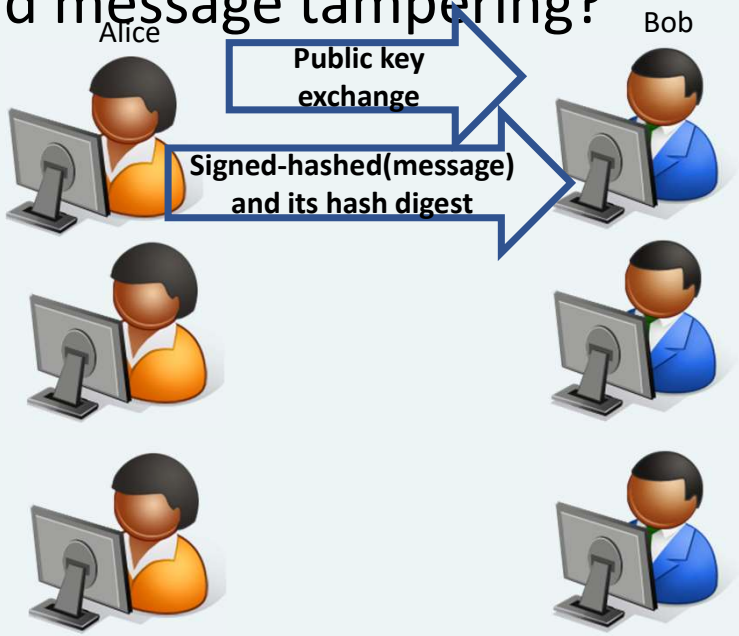
Conclusion about message tampering

- **We can use ‘strong’ hashing algorithms against message tampering.**

A combination of digital signature + hashing against impersonation and message tampering

- impersonation and message tampering?

1. Alice generates her pair public-private keys
2. Alice hashes a message with a hashing algorithm and encrypts it with her private key



3. Bob decrypts the message with Alice's public key
→ If it works: the message came from Alice
4. Bob hashes the received message with the same hashing algorithm as Alice
5. Bob compares the hash digest he computed with the one received from Alice
→ If they match: the message has not been tampered with

Conclusion

- We have seen 4 threats which affect client-server communication over the insecure Internet

Eavesdropping/snooping – confidentiality problem: encryption
Impersonation – authentication problem: digital signature
Replay – integrity problem: digital signature + timestamp
Tampering – integrity problem: hashing

- Combinations of solutions can deal with more than one threat

Further Reading

- *Computer Networking: A Top Down Approach* , 4th edition. Jim Kurose and Keith Ross. Addison-Wesley, July 2007.
- Security in Computing, 3rd edition. Charles P. Pfleeger and Shari L. Pfleeger. Pearson Education, Inc. 2003.
- Message Replay Detection: http://msdn.microsoft.com/en-us/library/ff649371.aspx
- Wang, X., Feng, D., Lai, X. and Yu, H. (2004). 'Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD', Cryptology ePrint Archive, Report 2004/199, available at http://eprint.iacr.org/2004/199
- Xiaoyun Wang , Yiqun Lisa Yin , Hongbo Yu (2005). Finding Collisions in the Full SHA-1. CRYPTO'05. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.4261
- https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

Sensitivity: Internal