# Security

3. Digital Certificates & Transport Layer Security

# Digital Certificates

# (Public) Key Management

- **Q.** When you receive a public key, how can you be sure that it is authentic?

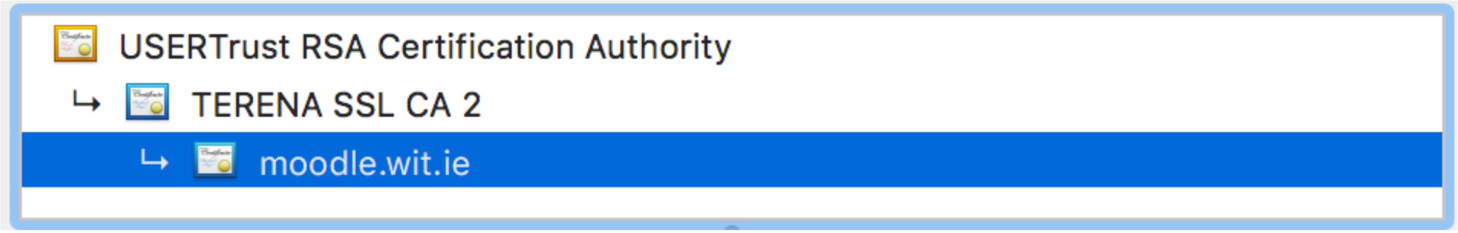
- **A.** If the received public key is **digitally signed** by someone whose own public key you have and are sure is correct **and** you trust them to sign keys responsibly.

# Digital Certificate – components

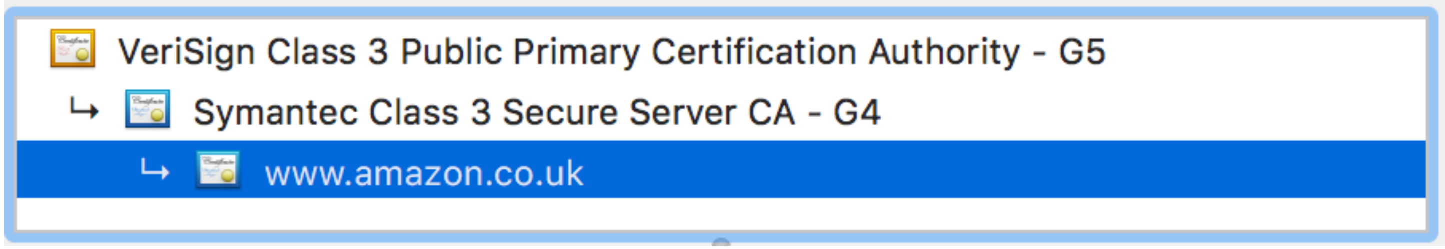
- Most important components of a digital certificate:
  - Subject (owner)
    - The name on the certificate – i.e. to whom it was issued
  - Subject’s public key
    - The purpose of a certificate is to validate the public key of the subject
  - Issuer (Certificate Authority)
    - The identity of entity that signed the certificate
  - Issuer’s digital signature
  - Serial number
    - Unique identifier for checking against revocation lists
  - Validity period
    - Start date; expiry date

# Chain of trust

- Can build up a chain of trusts with linked digital certificates
- This is the basis of what are known as Public Key Infrastructures (PKIs)

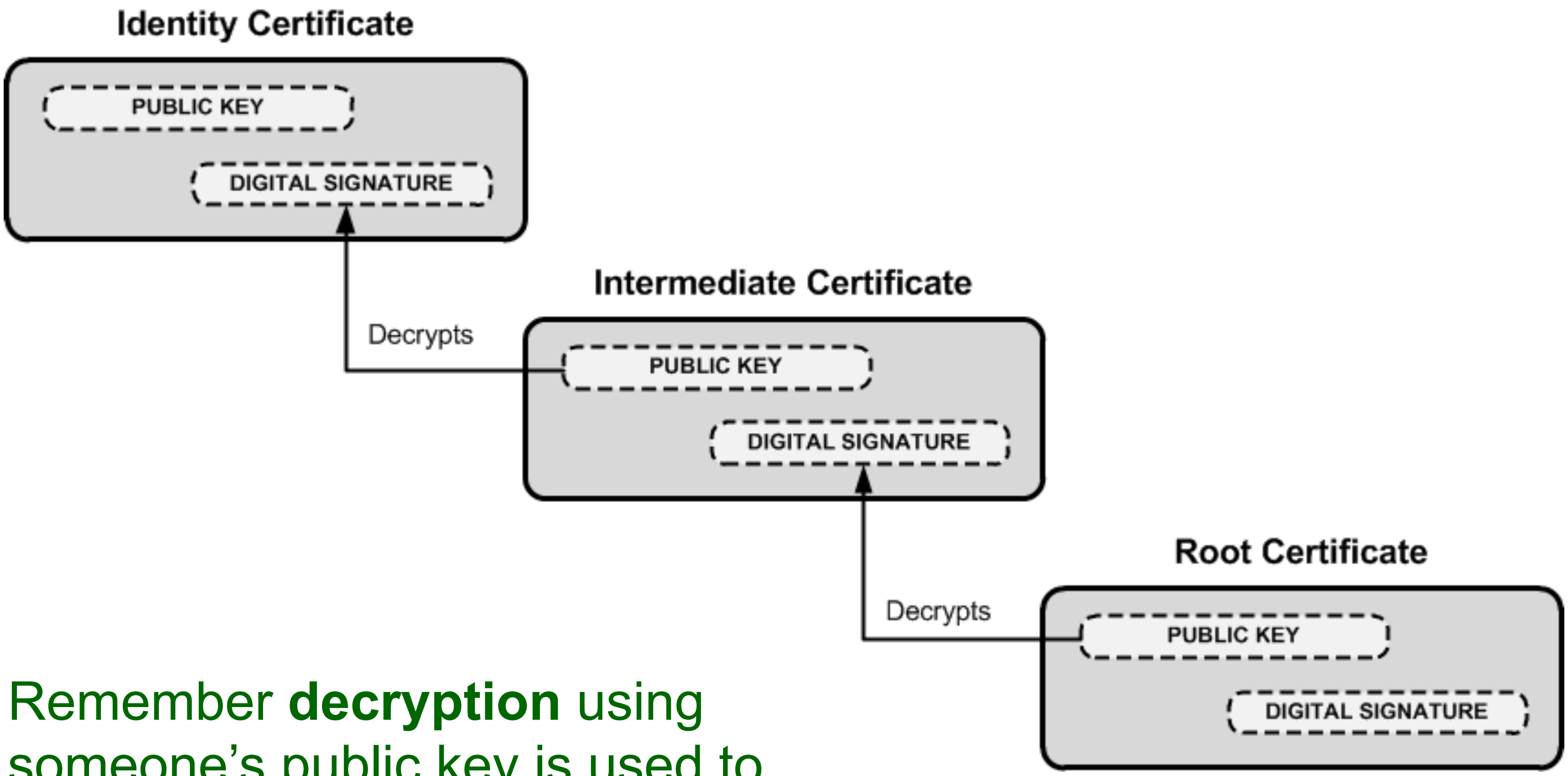


USERTrust RSA Certification Authority
↳ TERENA SSL CA 2
↳ moodle.wit.ie



VeriSign Class 3 Public Primary Certification Authority - G5
↳ Symantec Class 3 Secure Server CA - G4
↳ www.amazon.co.uk

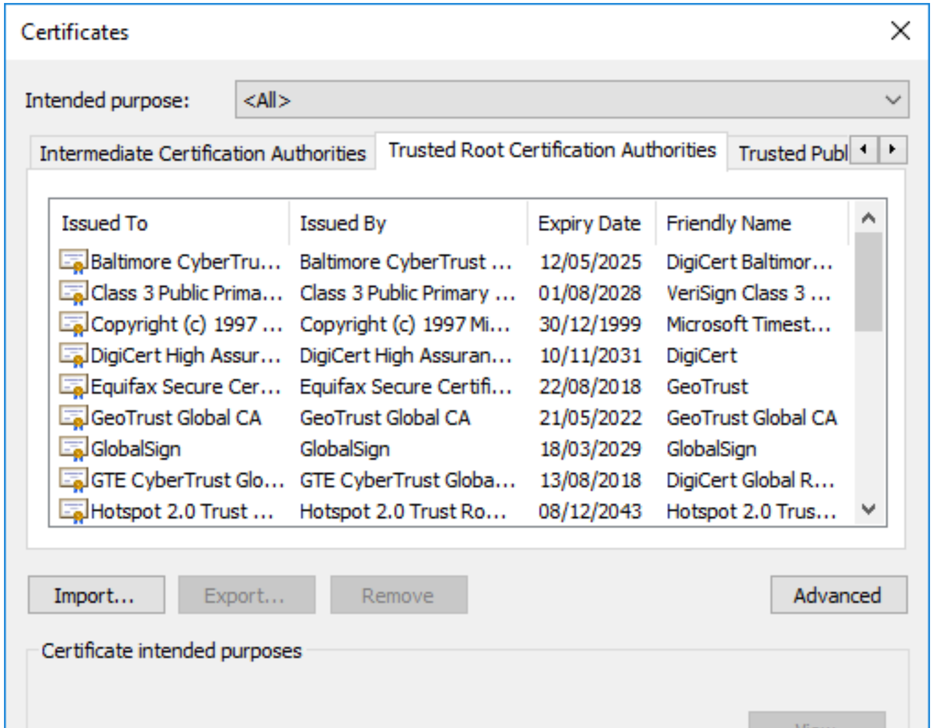
# Verification using chain of trusts



Remember **decryption** using someone’s public key is used to **verify** their signature

# Chain of trust

- The buck must stop somewhere. Ultimately, at the end of the chain, you must trust a public key that is not signed (usually belonging to some recognised “authority”).
  - In your browser, this is one of the trusted root certificate authorities



Certificates
Intermediate Certification Authorities
Trusted Root Certification Authorities
Trusted Publ
Import...
Export...
Remove
Advanced
Certificate intended purposes

Intended purpose: <All>

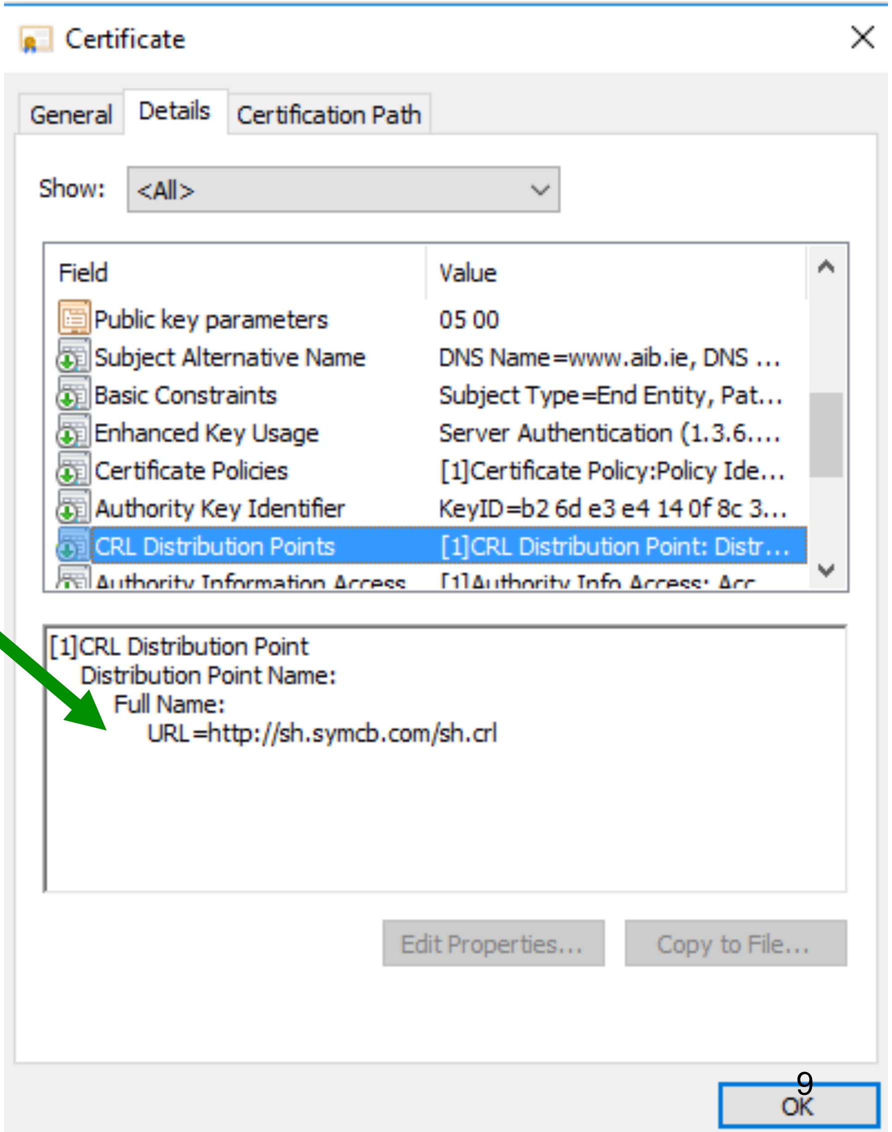
| Issued To | Issued By | Expiry Date | Friendly Name |
|---|---|---|---|
| Baltimore CyberTru... | Baltimore CyberTrust ... | 12/05/2025 | DigiCert Baltimor... |
| Class 3 Public Prima... | Class 3 Public Primary ... | 01/08/2028 | VeriSign Class 3 ... |
| Copyright (c) 1997 ... | Copyright (c) 1997 Mi... | 30/12/1999 | Microsoft Timest... |
| DigiCert High Assur... | DigiCert High Assuran... | 10/11/2031 | DigiCert |
| Equifax Secure Cer... | Equifax Secure Certifi... | 22/08/2018 | GeoTrust |
| GeoTrust Global CA | GeoTrust Global CA | 21/05/2022 | GeoTrust Global CA |
| GlobalSign | GlobalSign | 18/03/2029 | GlobalSign |
| GTE CyberTrust Glo... | GTE CyberTrust Globa... | 13/08/2018 | DigiCert Global R... |
| Hotspot 2.0 Trust ... | Hotspot 2.0 Trust Ro... | 08/12/2043 | Hotspot 2.0 Trus... |

# Certificate Expiry & Revocation

- A Digital Certificate doesn’t last for ever
- It normally **expires** after a certain time and must be renewed
- It may be **revoked:**
  - If the subject’s private key is compromised
  - If there is a change in status of the subject
  - If the CA’s private key is compromised
- Revoked Certificates are placed on a Certificate Revocation List (CRL)

# Certificate Revocation

- An issue is where to find CRL to check if cert has been revoked
  - One solution is to provide as part of certificate URL pointing to CRL
  - Another solution is OCSP (online certificate status protocol) which allows real time queries.
  - Another is to just rely on local list which is refreshed by browser updates (Chrome does this)



Certificate

Show: <All>

| Field | Value |
|---|---|
| Public key parameters | 05 00 |
| Subject Alternative Name | DNS Name=www.aib.ie, DNS ... |
| Basic Constraints | Subject Type=End Entity, Pat... |
| Enhanced Key Usage | Server Authentication (1.3.6.... |
| Certificate Policies | [1]Certificate Policy:Policy Ide... |
| Authority Key Identifier | KeyID=b2 6d e3 e4 14 0f 8c 3... |
| CRL Distribution Points | [1]CRL Distribution Point: Distr... |
| Authority Information Access | [1]Authority Info Access: Acc |

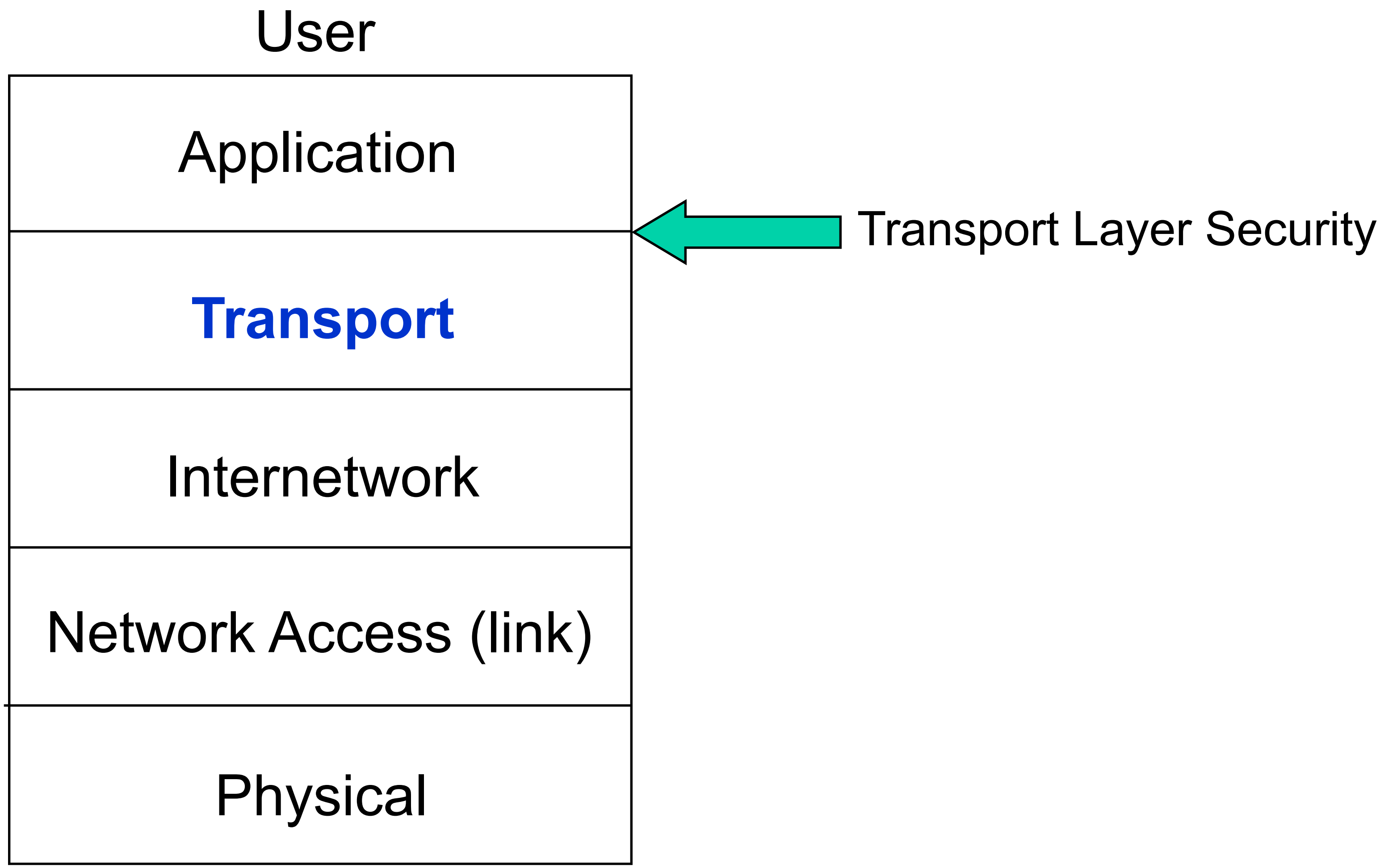
```
[1]CRL Distribution Point
   Distribution Point Name:
      Full Name:
         URL=http://sh.symcb.com/sh.crl
```

# Transport Layer Security

# Securing Web content

- HTTP by itself doesn’t provide any security
- The approach to securing web content is to:
  - Leave HTTP as it is
  - Add security just above the transport layer
- This has been variously known as
  - Secure Socket Layer (SSL)
    - Originated by Netscape
  - Transport Layer Security (TLS)
    - Vendor-neutral standard
    - RFC 5246 (TLS 1.2)

# Reminder: TCP/IP (Internet) Protocol Stack



User

Application

**Transport**

Internetwork

Network Access (link)

Physical

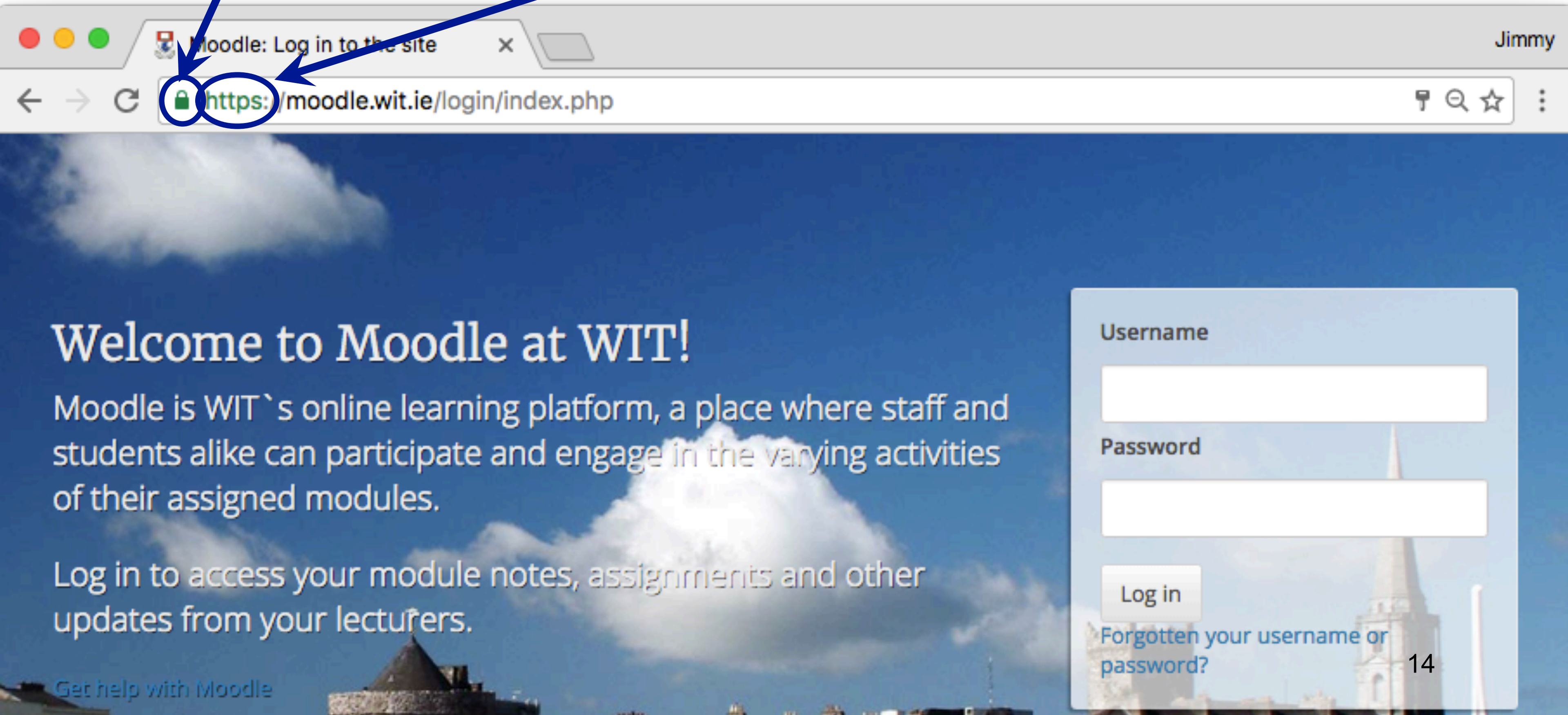
Transport Layer Security

# TLS Requirements

- Client contacts Server (possibly for the first time)
  - Spontaneity
- Client conveys secret info to Server
  - Confidentiality
- Authentication − Who’s on the other side?
  - Server Authentication – required
  - *Client authentication – optional*
- User doesn’t not want to know about security
  - Transparency
    - This property means that other protocols can also work over SSL/TLS (it’s not tied to HTTP)

# Recognising a TLS (SSL) page

- Closed lock symbol
- URL begins with http**s**: rather than http:



Moodle: Log in to the site
Jimmy
https://moodle.wit.ie/login/index.php

Welcome to Moodle at WIT!

Moodle is WIT`s online learning platform, a place where staff and students alike can participate and engage in the varying activities of their assigned modules.

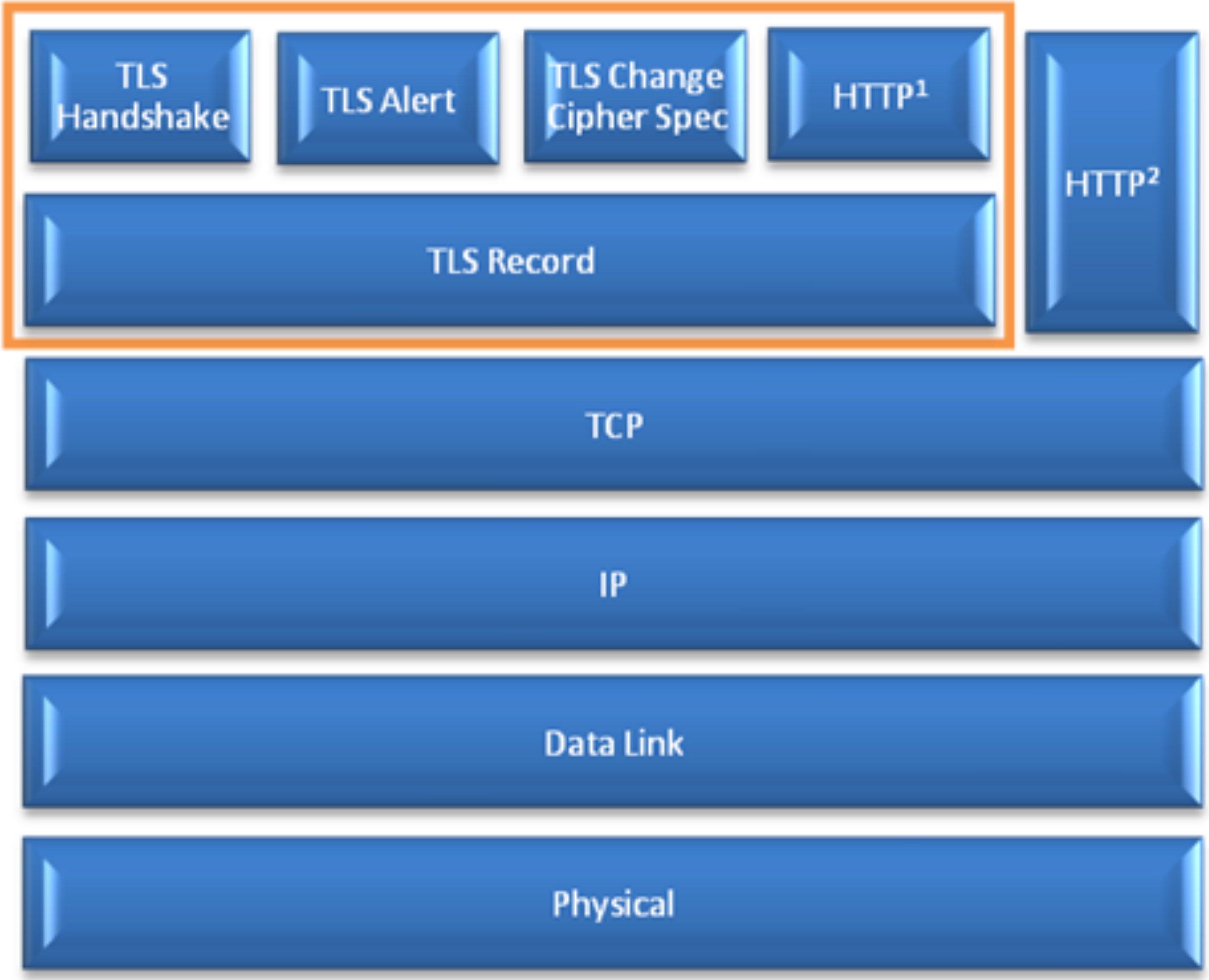
Log in to access your module notes, assignments and other updates from your lecturers.

Get help with Moodle
Username
Password
Log in
Forgotten your username or password?

# TLS (SSL) Protocol Overview

- TLS (SSL) has 2 layers of protocols:
- One layer is a set of protocols for setting up a session, changing parameters, etc
  - TLS Handshake Protocol
  - TLS Change Cipher Spec Protocol
  - TLS Alert Protocol
- The other is the “workhorse”, doing the encryption and authentication
  - TLS Record Protocol

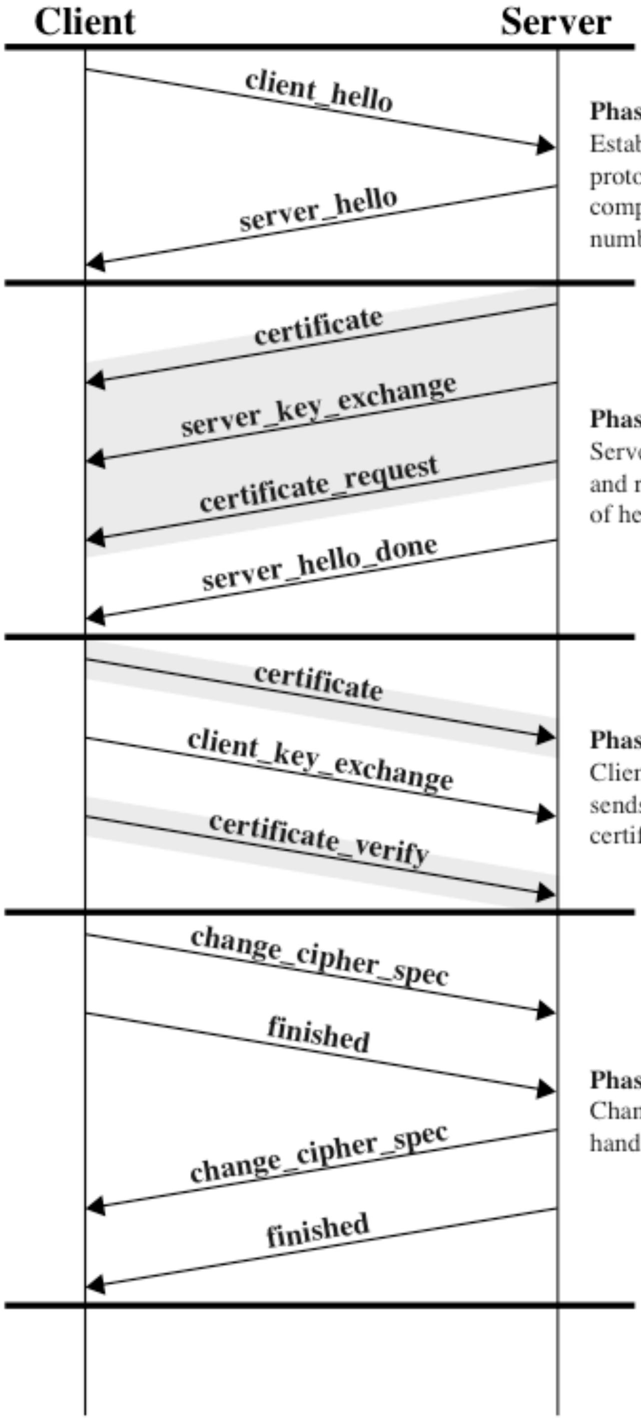
# TLS Architecture



# TLS Handshake Protocol

1. Agree TLS/SSL version & **cipher suite** (algorithms and settings)
2. Client authenticates server using its certificate; server optionally authenticates client.
3. Client generates random session key and shares with server by encrypting it with server's public key (from its cert)

➢ Client and server can now communicate using shared session key (for symmetric encryption)



**Phase 1**
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

**Phase 2**
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

**Phase 3**
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

**Phase 4**
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

# TLS Cipher Suites and Alerts

- An official registry of cipher suites and alert types is maintained by the Internet Assigned Names and Numbers Authority (IANA)
  - http://www.iana.org/assignments/tls-parameters/

# Is TLS secure?

- So much relies on TLS nowadays that it is fair to ask whether it can be considered secure

- The answer is yes and no
  - Yes, the protocol seems to be secure if used correctly
  - However it is very fragile – any of a large number of conditions can break it (completely)

# Is TLS secure?

- TLS fails if:
  - One “bad” certificate authority is added to the client’s list of trusted CAs
  - One of the “good” CAs is careless or unlucky
  - Weak algorithms are used
  - Key generation is weak (often due to bad pseudo-random number generator)
  - One side tricks the other into “stepping down” to use a weak algorithm or key length (e.g. MD5, 512 bit RSA). This is possible as TLS allows the two sides to negotiate these.
  - Client doesn’t check for certificate revocation

# Is TLS secure?

- TLS fails if (continued):
  - Client or server is modified by malware
  - Client or server has software bugs (e.g. *Heartbleed*)
  - Client is modified by system administrator to use company’s CA list
  - Pages contain a mix of secure and insecure content (frames, images, etc)
  - Users fail to understand security warnings and/or are conditioned to ignore them
  - Server fails to renew certificates