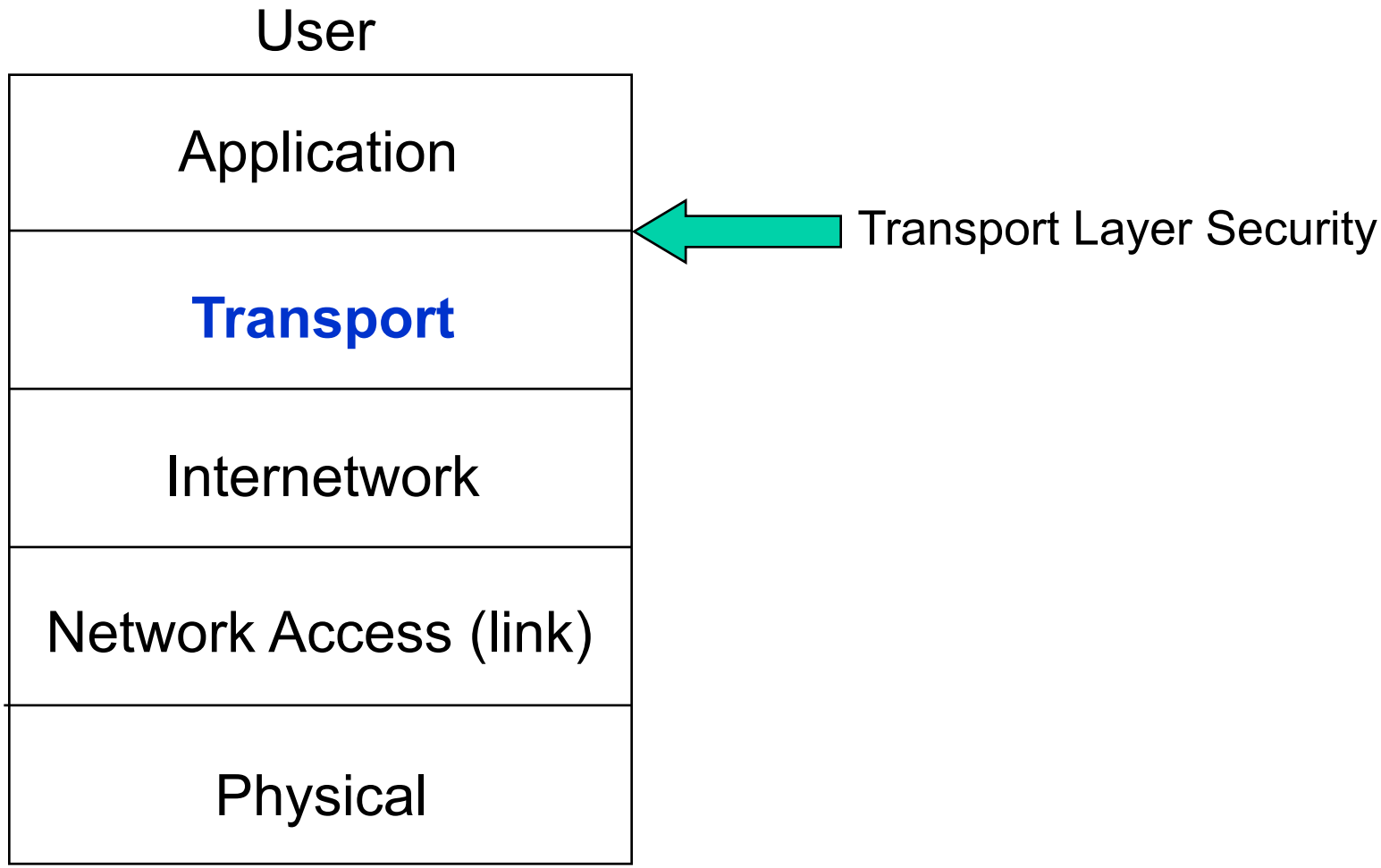
Security

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

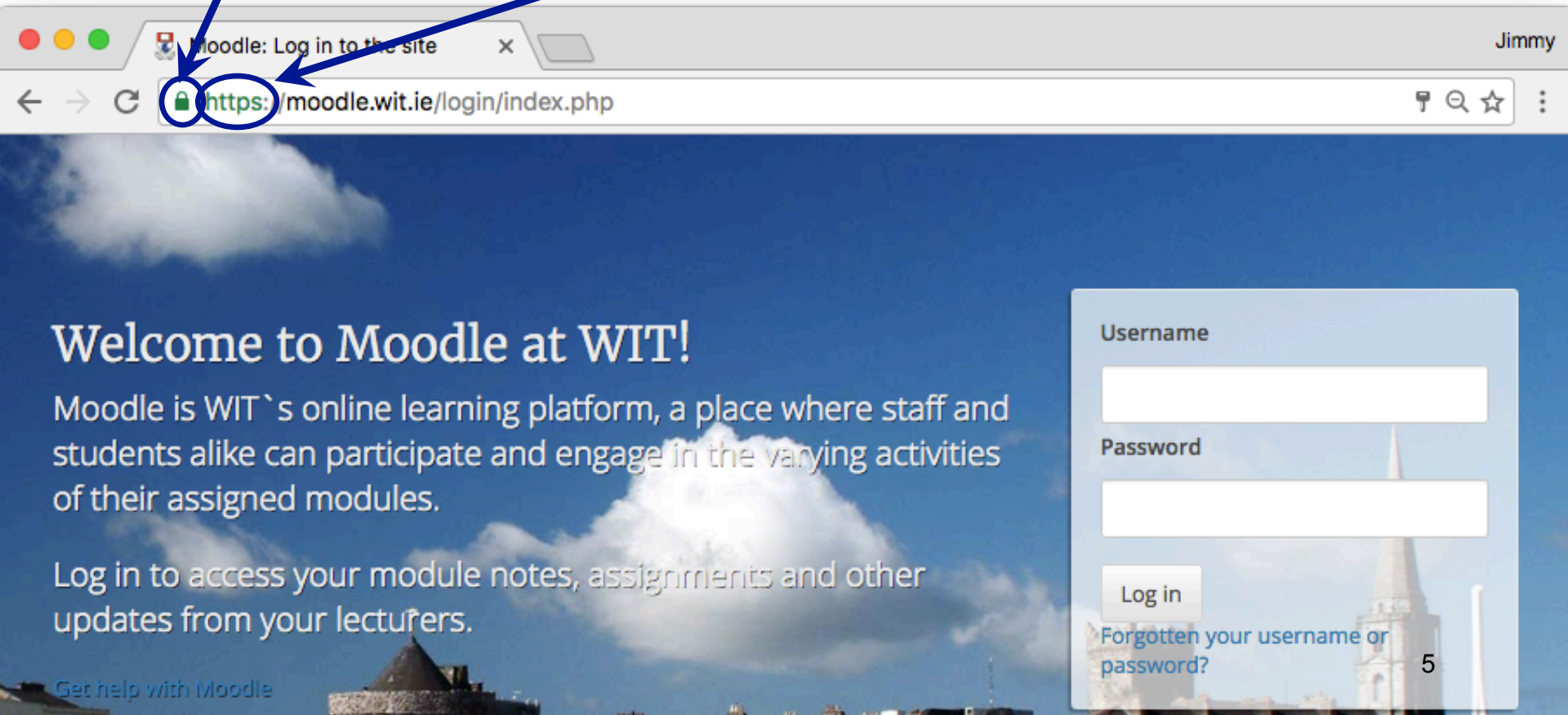


TLS Requirements

- 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 TLS (it's not tied to HTTP)

Recognising a TLS-secured page

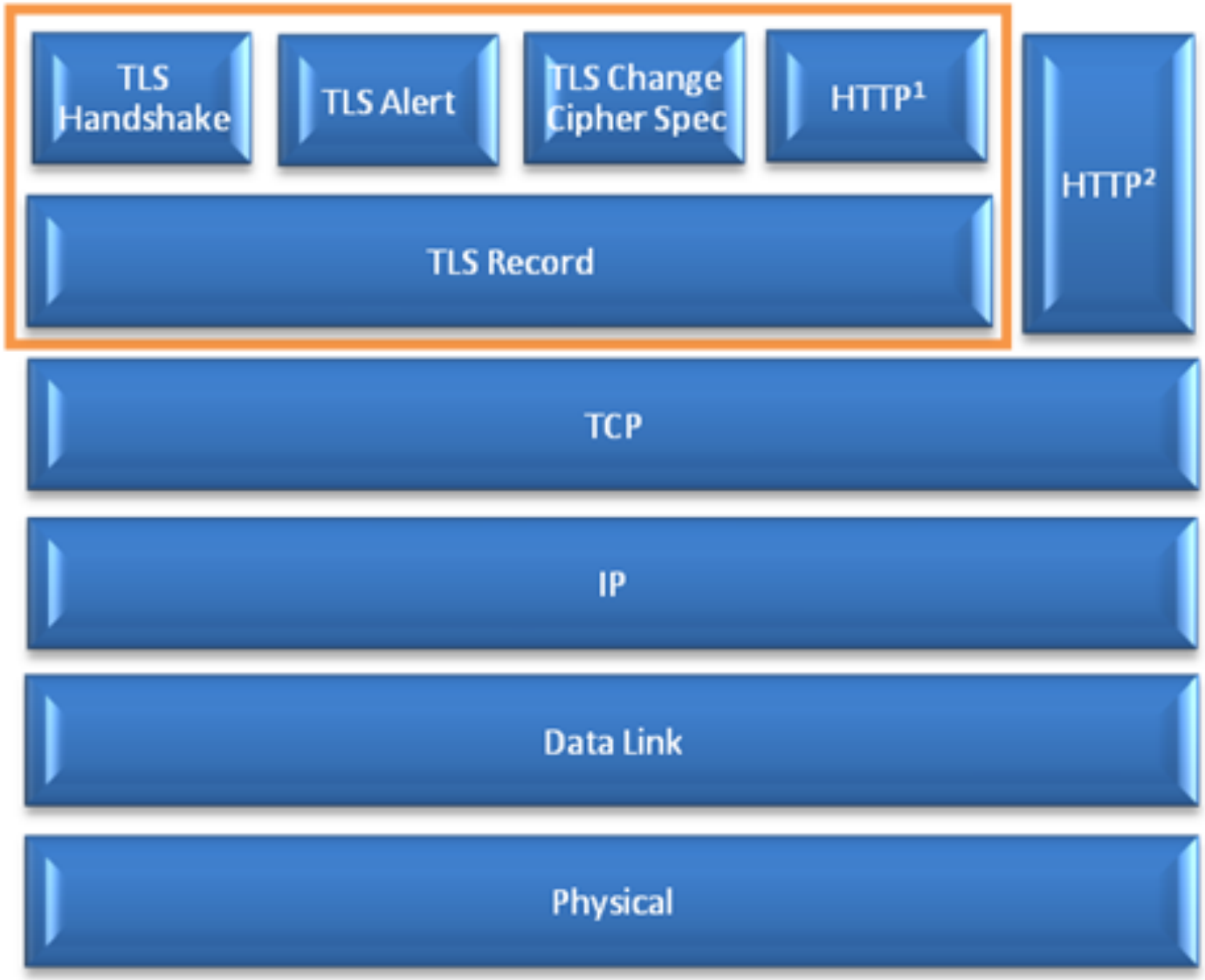
- Closed lock symbol
- URL begins with https**s**: rather than http:



TLS Protocol Overview

- TLS 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

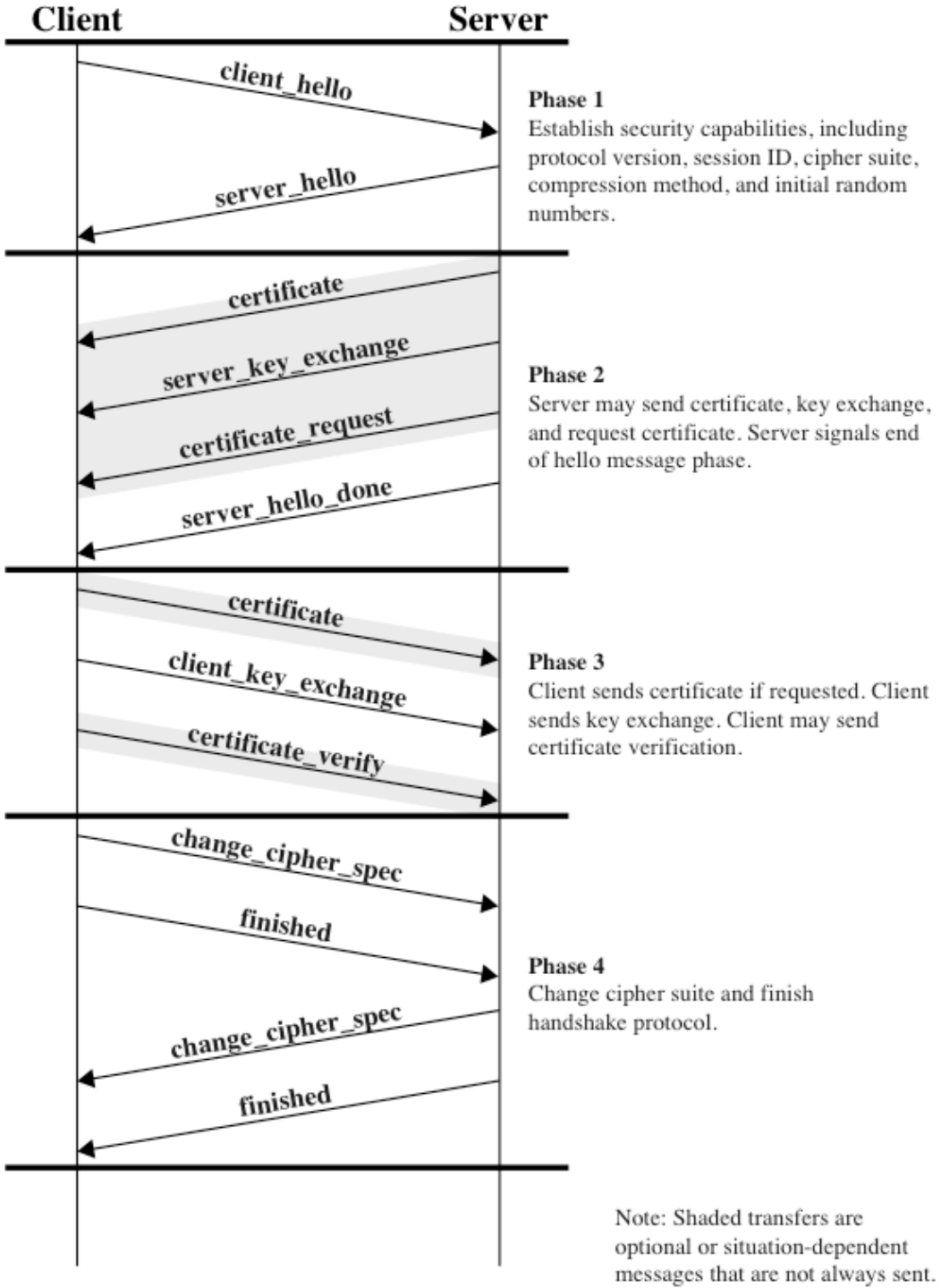
TLS Architecture



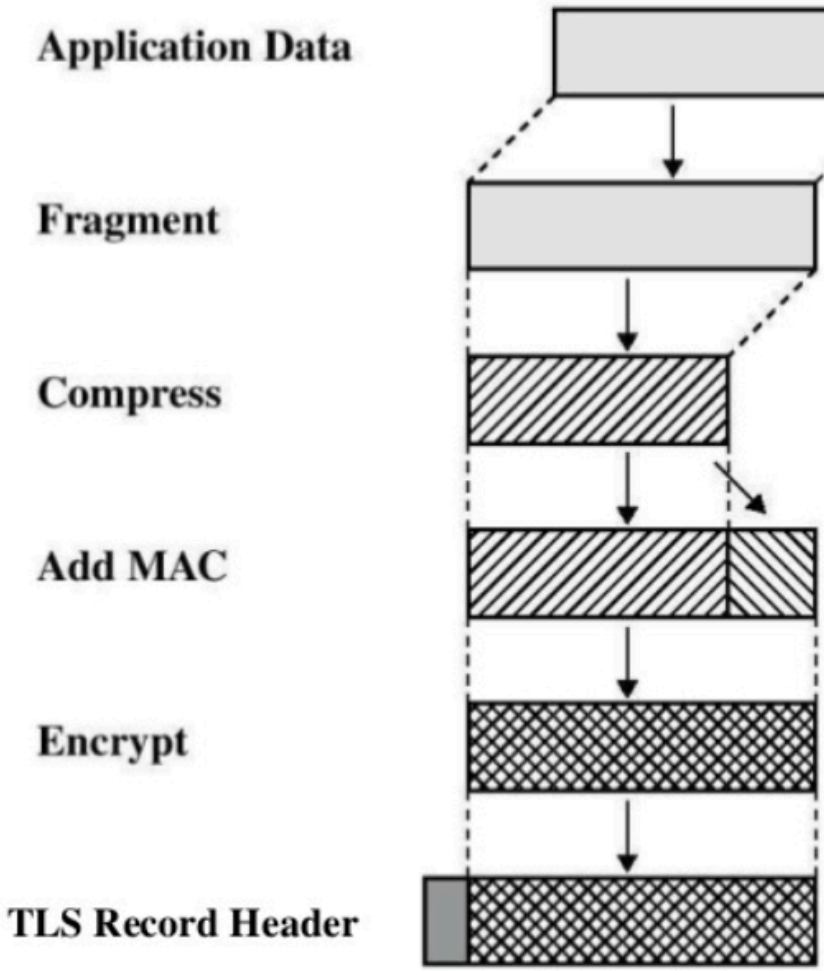
TLS Handshake Protocol

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)



TLS Record Protocol



Compression is an optional feature (often disabled due to vulnerability).

MAC-based authentication and data encryption uses shared secret key (exchanged using TLS Handshake Protocol).

TLS header added to secured fragment. This (header + payload) is then passed to the next layer (TCP).

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