SSL/TLS CONT.
Lecture 11a

COMPSCI 726
Network Defence and Countermeasures

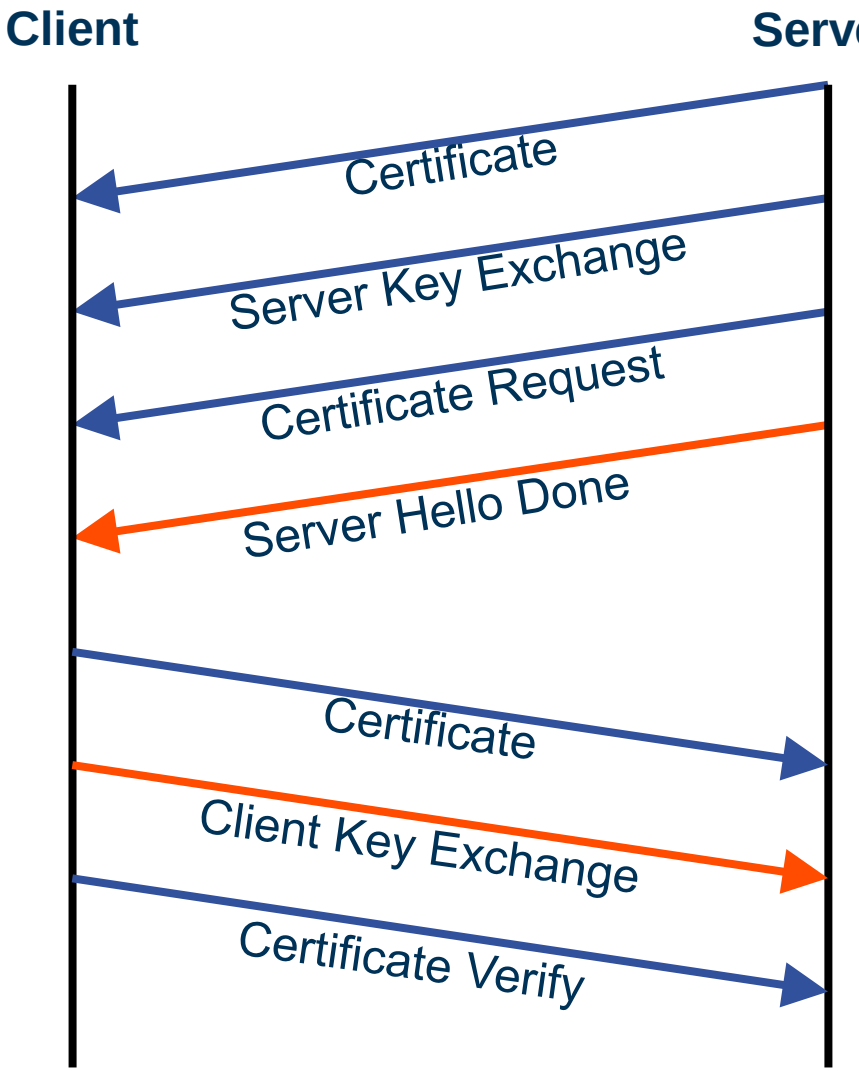
Muhammad **Rizwan** Asghar

August 11, 2021

Source of some slides: **University of Twente**

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

HANDSHAKE PROTOCOL: KEY EXCHANGE AND AUTHENTICATION



If requested by the client, the server sends its authentication data plus keys

If necessary, the client sends its authentication data plus keys

HANDSHAKE PROTOCOL: PHASE 2

- Certificate message
  - Server's X.509v3 certificate followed by optional chain of certificates
  - Required for RSA, Fixed DH, Ephemeral DH but not for Anonymous DH
- Server Key Exchange message
  - Not needed for RSA and Fixed DH
  - Required for Ephemeral DH and Anonymous DH
  - Needed for RSA where the server has signature-only key

HANDSHAKE PROTOCOL: PHASE 2

- Server Key Exchange message cont.
  - Signed by the server
  - Signature is on hash of
    - ClientHello.random, ServerHello.random
    - Server Key Exchange parameters
- Certificate Request message
  - Requests a certificate from the client
- Server Done message
  - Ends phase 2, always required

HANDSHAKE PROTOCOL: PHASE 3

- Certificate message
  - Send if the server has requested certificate and the client has appropriate certificate
    - Otherwise, send no_certificate alert
- Client Key Exchange message
  - Content depends on type of key exchange (see next slide)
- Certificate Verify message
  - Can be optionally sent following a client certificate with signing capability
  - Signs hash of master secret (established by key exchange) and all handshake messages so far
  - Provides evidence of possessing private key corresponding to the certificate

HANDSHAKE PROTOCOL: PHASE 3

- Client Key Exchange message
  - RSA
    - Client generates 48-byte pre-master secret, which is encrypted with the server's RSA public
  - Ephemeral or Anonymous DH
    - Client's public DH value
  - Fixed DH
    - Null
    - It uses public key previously sent in the Certificate message

HANDSHAKE PROTOCOL: PHASE 3

- 48-byte pre-master secret
  - RSA
    - Generated by the client
    - Sent encrypted to the server
  - DH
    - Both sides compute the same value
    - Each side uses its own private value and the other side's public value

HANDSHAKE PROTOCOL: PHASE 3

```
pre_master_secret: 48 bytes


master_secret = PRF(pre_master_secret,
                    "master secret",
                    ClientHello.random,
                    ServerHello.random)
```

PRE-MASTER SECRET AND MASTER SECRET

- Pre_master_secret
  - Computed for each session during handshaking
  - Could be the same for all sessions
  - It is not stored (so safe)

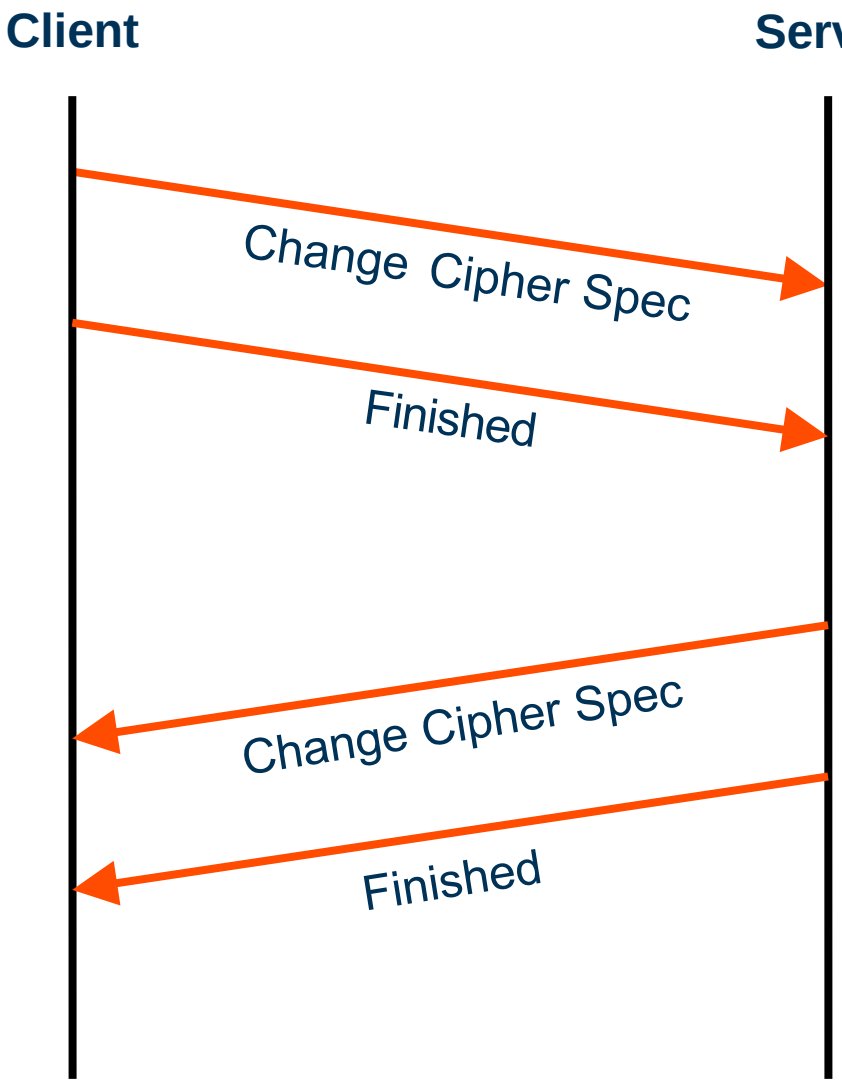
- Master_secret
  - It is stored for the lifetime of the session
  - May be compromised
  - If compromised, damage limited to a session

GENERATION OF CRYPTOGRAPHIC PARAMETERS

- An SSL session needs four keys
  - Four keys
    - Client write MAC secret
    - Server write MAC secret
    - Client write key (for encryption)
    - Server write key
  - Two IVs
    - Client write IV (for CBC)
    - Server write IV

- These parameters are generated from the master secret

CHANGE CIPHER SPEC PROTOCOL

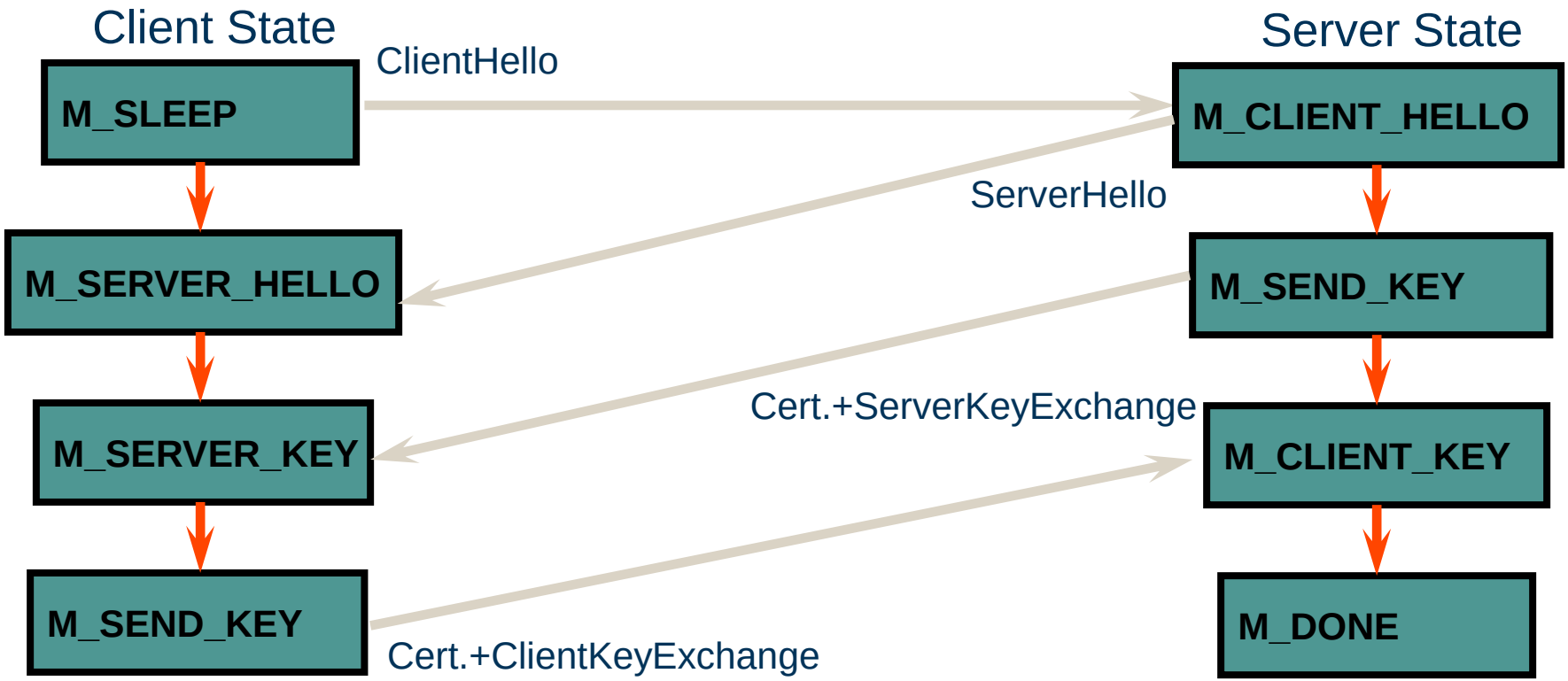


- `Change cipher spec` activates new algorithms
- `Finished` verifies new algorithms
- After the server sends `Finished`, actual data exchange starts

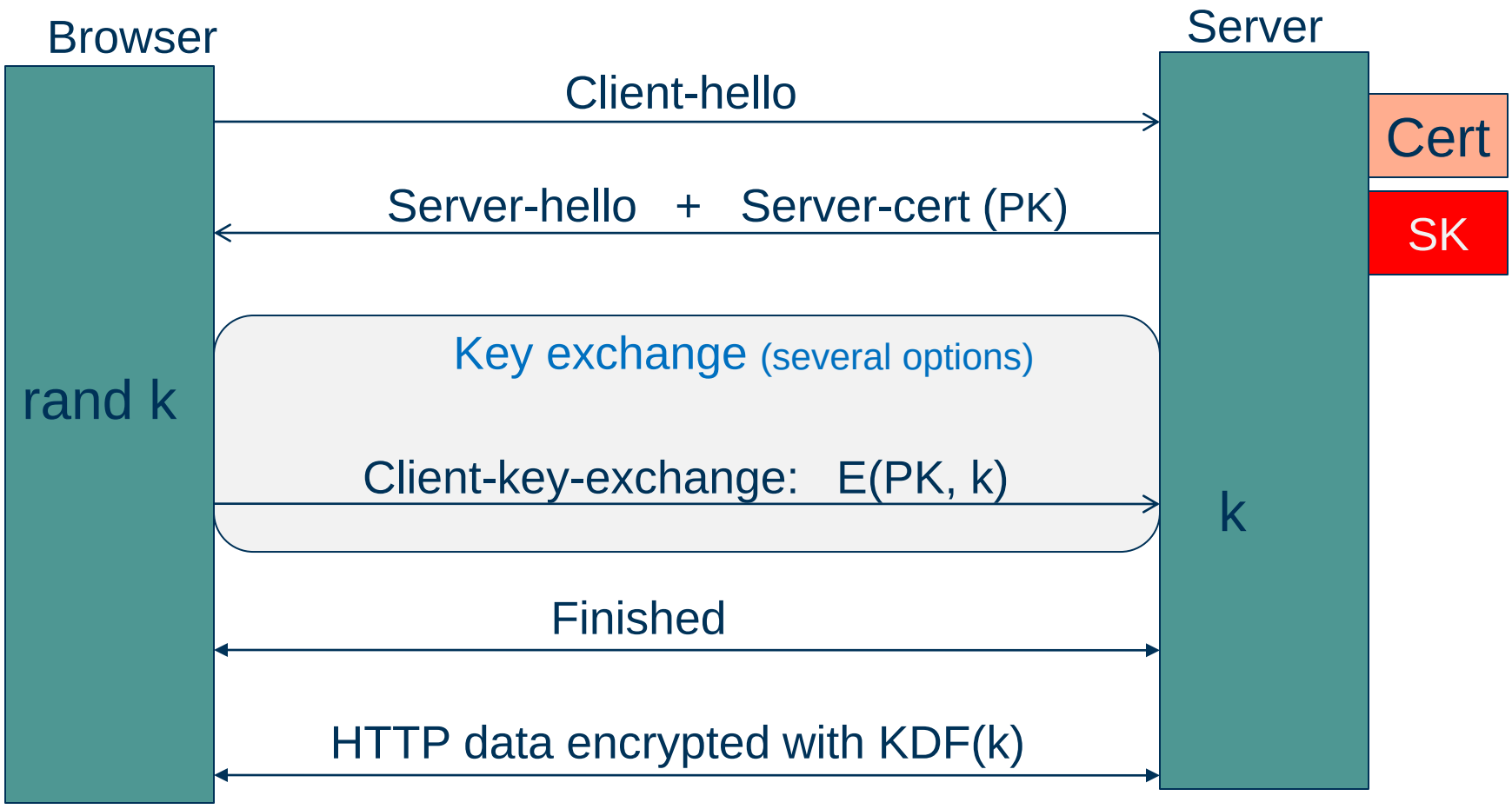
HANDSHAKE PROTOCOL: PHASE 4

- Change Cipher Spec message
  - Not considered part of the handshake protocol but in some sense is part of it
- Finished message
  - Sent under new algorithms and keys
  - Content is hash of all previous messages and master secret

PARTICIPANTS AS FINITE-STATE MACHINES



PROTOCOL IN ACTION



Most common: Server authentication only

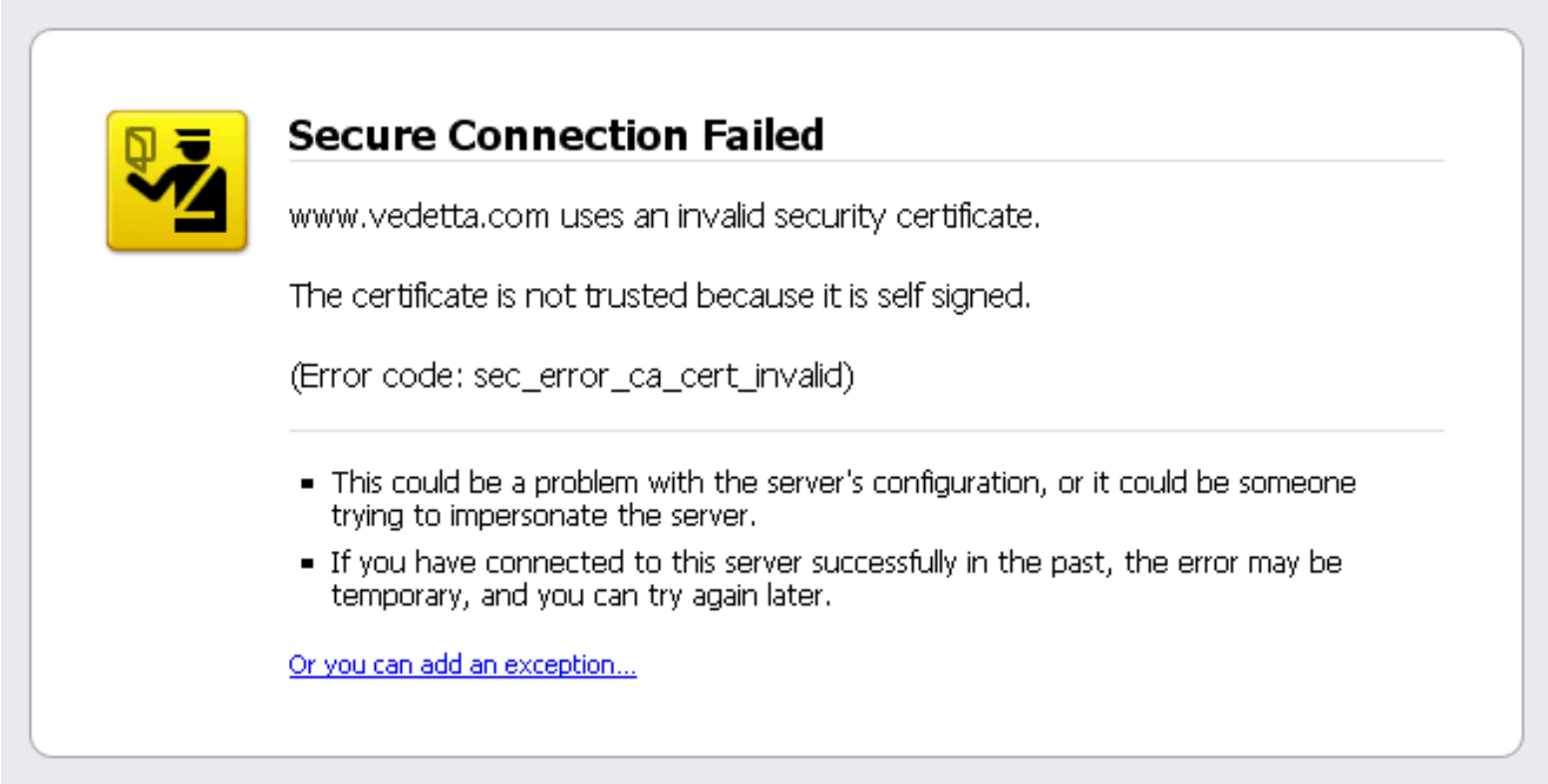
SSL ALERT PROTOCOL

- Conveys SSL-related alerts to peer entities
- 2-byte alert messages
  - 1-byte level
    - Fatal or warning
  - 1-byte code
    - Alert code
- Severity
  - Fatal
  - Warning

ALERT PROTOCOL

- Specific alert
  - **Fatal:** Unexpected message, bad record MAC, decompression failure, handshake failure, and illegal parameter
  - **Warning:** Close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, and certificate unknown
- Compressed and encrypted like all SSL data

ALERT PROTOCOL



SSL ALERT MESSAGES

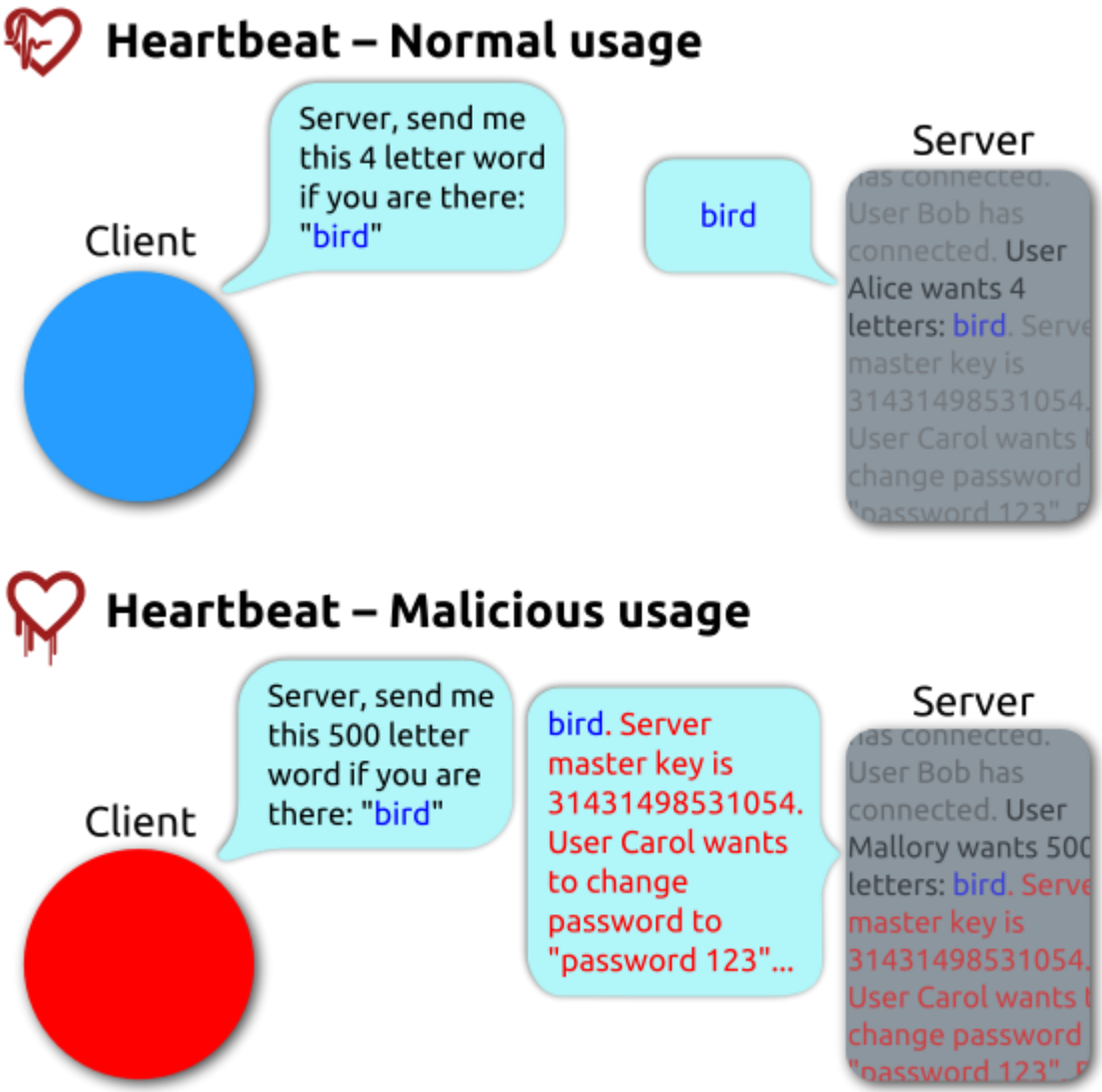
```
Warning or fatal

            close_notify(0),
            unexpected_message(10),
            bad_record_mac(20),
            decryption_failed(21),
            record_overflow(22),
            decompression_failure(30),
            handshake_failure(40),
            bad_certificate(42),
            unsupported_certificate(43),
            certificate_revoked(44),
            certificate_expired(45),
            certificate_unknown(46),
            illegal_parameter(47),
            unknown_ca(48),
            access_denied(49),
            decode_error(50),
            decrypt_error(51),
            export_restriction(60),
            protocol_version(70),
            insufficient_security(71),
            internal_error(80),
            user_canceled(90),
            no_renegotiation(100),
```

HEARTBEAT PROTOCOL

- A periodic signal generated to indicate normal operation or to **synchronise** other parts of a system
- Typically used to monitor the availability of a protocol entity
- Defined in 2012 in RFC 6250
- Runs on top of the TLS Record Protocol
- Use is established during Phase 1 of the handshake protocol
- Each peer indicates whether it supports heartbeat
- Serves two purposes:
  - Assures the sender that the **recipient is still alive**
  - Generates activity across the connection during idle periods

HEARTBLEED



For this image, thanks to FenixFeather

APPLICATION PORTS

- https 443
- ssmtp 465
- snntp 563
- sldap 636
- spop3 995
- ftp-data 889
- ftps 990
- imaps 991
- telnets 992
- ircs 993

SUMMARY

- SSL/TLS is based on two main sub-protocols
  - Handshake protocol for key establishment
  - Record protocol for secure communication
- It uses authenticated encryption
  - Variation: MAC and then encrypt
- Key exchange methods use RSA and DH
- A master secret is generated from a pre-master secret for providing forward secrecy

RESOURCES

- Read Chapter 5 of
  Network Security Essentials – Applications and Standards
  Fourth Edition
  William Stallings
  Prentice Hall
  ISBN 0-13-706792-5

- OpenSSL: https://www.openssl.org
  - A free open-source implementation of SSL

RESOURCES (2)

- Georgiev, Martin, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, and Vitaly Shmatikov, **The most dangerous code in the world: validating SSL certificates in non-browser software**, In Proceedings of the ACM Conference on Computer and Communications Security, pp. 38-49. ACM, 2012, available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.296.556&rep=rep1&type=pdf

Questions?

Thanks for your attention!