SSL/TLS
Lecture 10

COMPSCI 726
Network Defence and Countermeasures

Muhammad **Rizwan** Asghar

August 9, 2021

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

THE UNIVERSITY OF AUCKLAND
NEW ZEALAND

WHAT IS SECURE SOCKET LAYER (SSL)?

- A protocol originally developed by Netscape
- Subsequently became Internet standard known as Transport Layer Security (TLS)
- Operates between TCP and application protocols
- The primary goal is to establish a **secure channel** between two communicating parties
- It is built into all major browsers and web servers

HISTORY OF SSL

- SSL 1.0
  - Internal Netscape design, early 1994
  - First version
- SSL 2.0
  - Published by Netscape, November 1994
  - Badly broken
- SSL 3.0
  - Designed by Netscape and Paul Kocher, November 1996

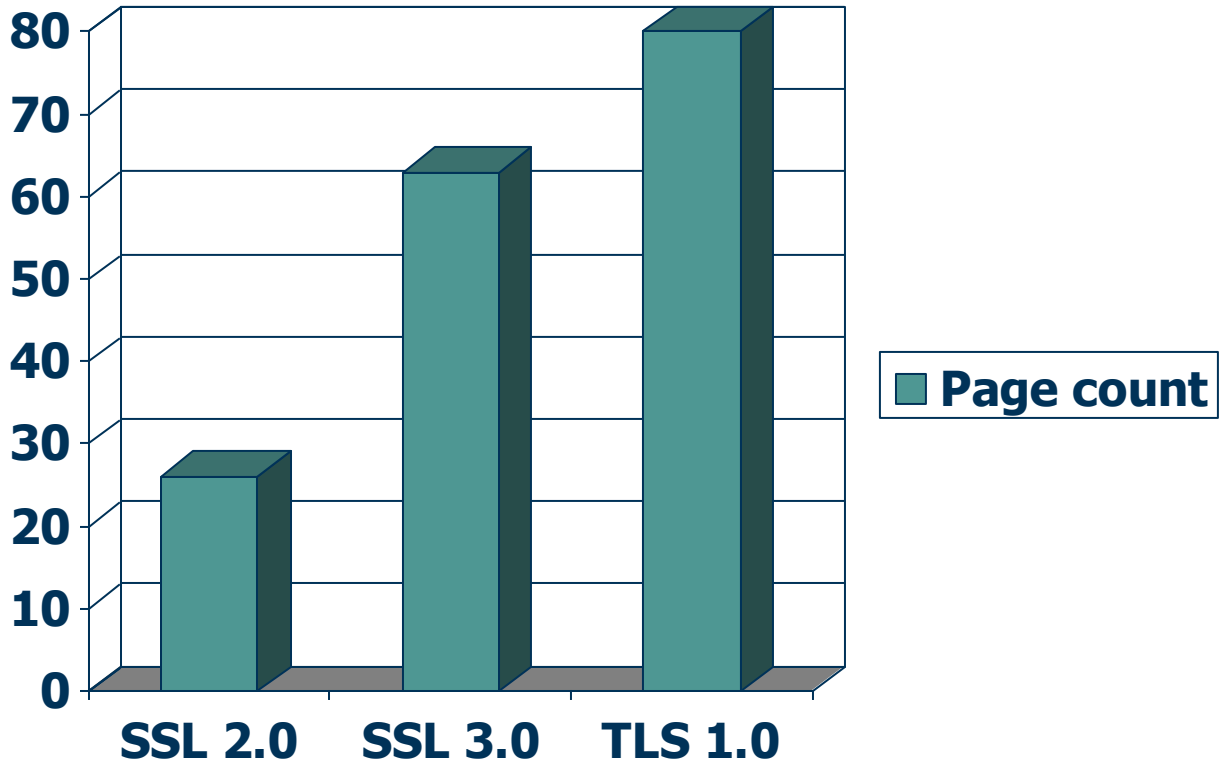
HISTORY OF TLS

- TLS 1.0
  - Internet standard based on SSL 3.0, January 1999
  - Not interoperable with SSL 3.0
- TLS 1.1
  - RFC 4346 in April 2006
  - Explicit IV and fixing the padding procedure
- TLS 1.2
  - RFC 5246 in August 2008 and RFC 6176 in March 2011
  - Adding cipher algorithms (e.g., AES)
- TLS 1.3
  - RFC 8446 in August 2018

REQUEST FOR COMMENTS (RFC)

- Network protocols are usually disseminated in the form of an RFC
- Intended to be a self-contained definition of the protocol
  - Describes the protocol in detail for those who will be implementing it or doing protocol analysis
  - Mixture of informal prose and pseudo-code
- Read some RFCs to get an idea of how protocols look like
- TLS version 1.0 is described in RFC 2246

EVOLUTION OF SSL/TLS RFC



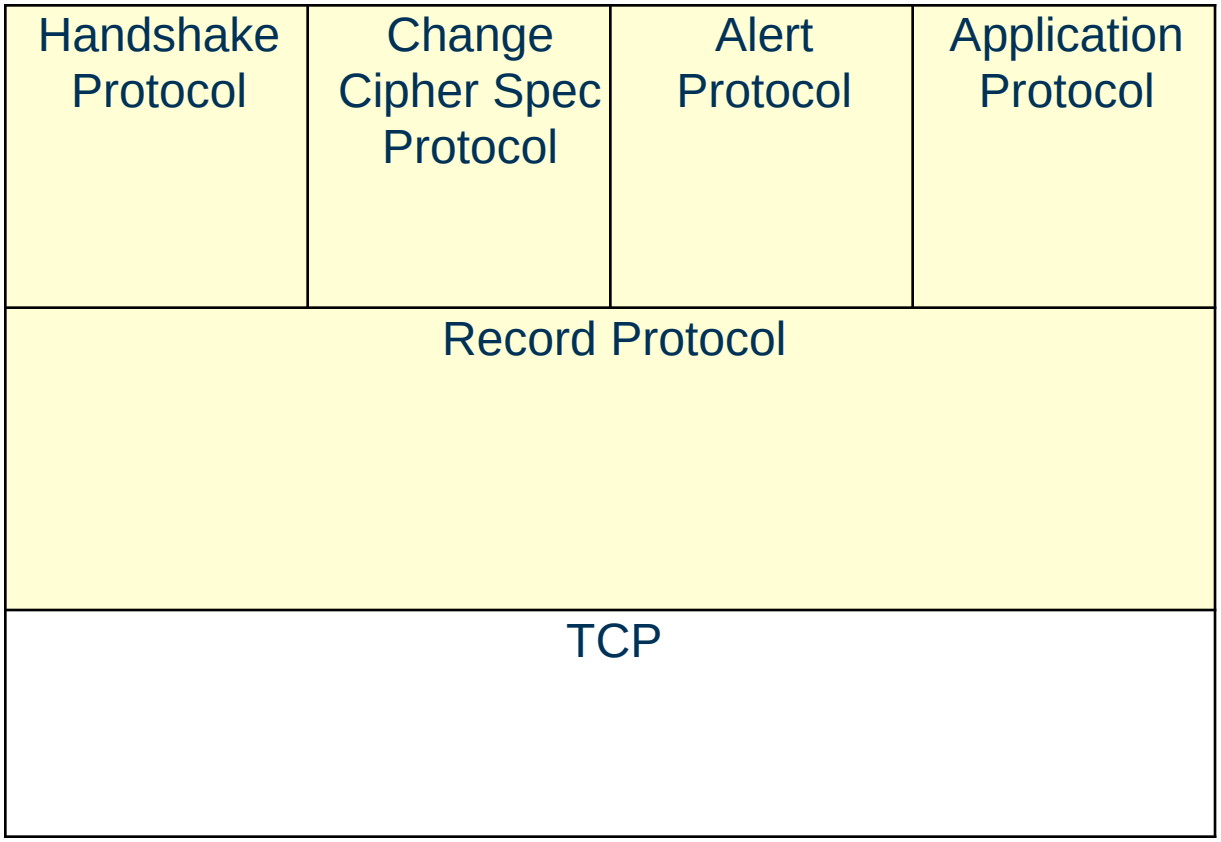
PROTOCOL GOALS

- Defined in RFC 4346
  - Cryptographic security
  - Interoperability
  - Extensibility
  - Relative efficiency
    - Session caching

PROTOCOL SERVICES

- Server authentication
- Client authentication
- Data confidentiality (encrypted connection)
  - This protects against eavesdroppers
- Data integrity
  - This protects against modifications
- It does not provide non-repudiation
- Generation/distribution of session keys
  - Integrated into the protocol
- Security parameter negotiation
- Compression and decompression

PROTOCOL ARCHITECTURE



Client / Server authentication
Algorithm and key negotiation

Confidentiality
Message Integrity

PROTOCOL BASICS

- There are two core sub-protocols
- Handshake protocol
  - Using public key cryptography to establish a shared secret key between the client and the server
- Record protocol
  - Use the secret key established in the handshake protocol to protect communication between the client and the server

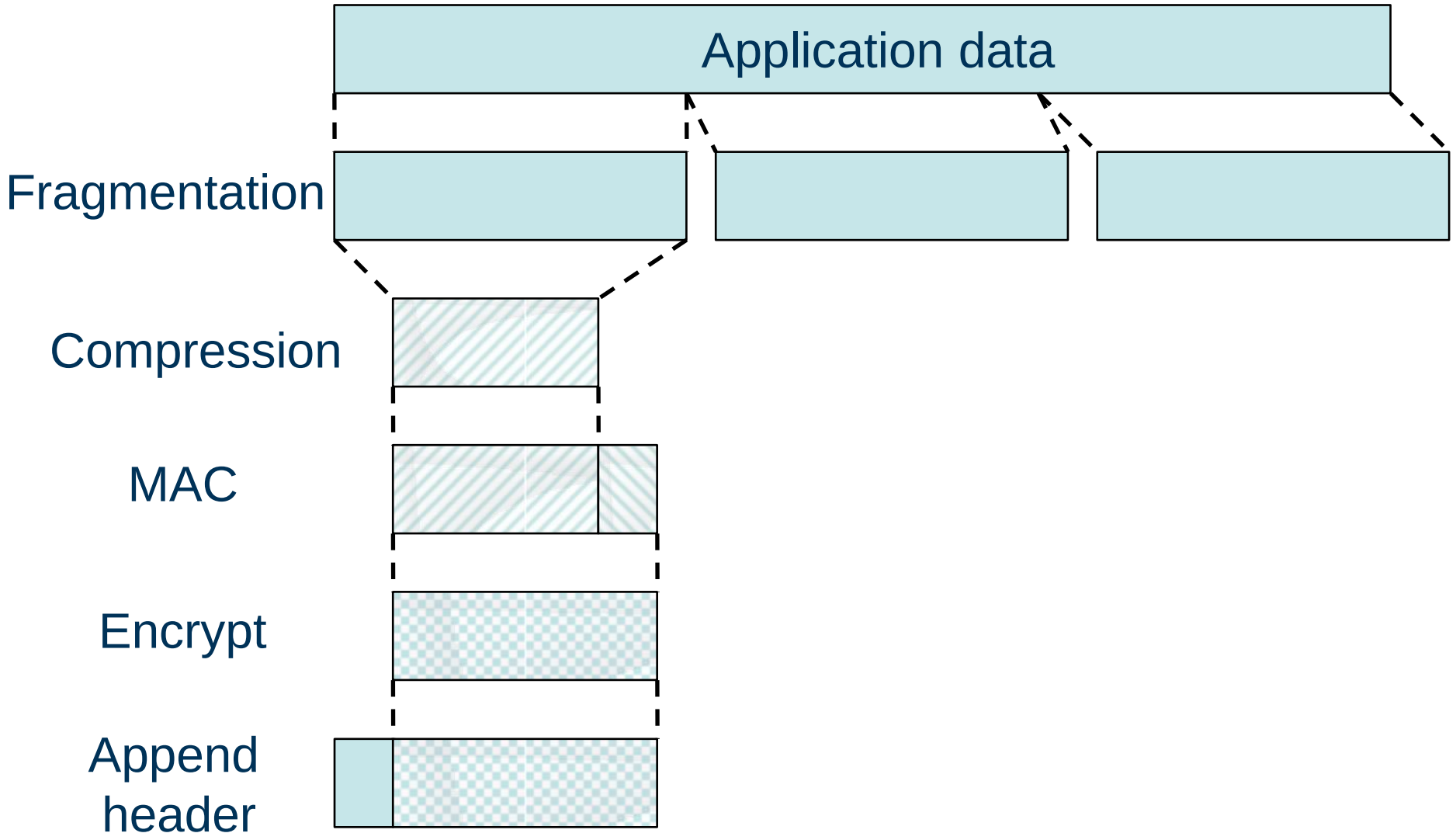
HANDSHAKE PROTOCOL

- Two parties: A client and a server
- Negotiate version of the protocol and the set of cryptographic algorithms to be used
  - Interoperability between different implementations of the protocol
- Authenticate client and server
  - E.g., using digital certificates to learn each other's public keys and verify each other's identity
- Use public keys to establish a shared secret

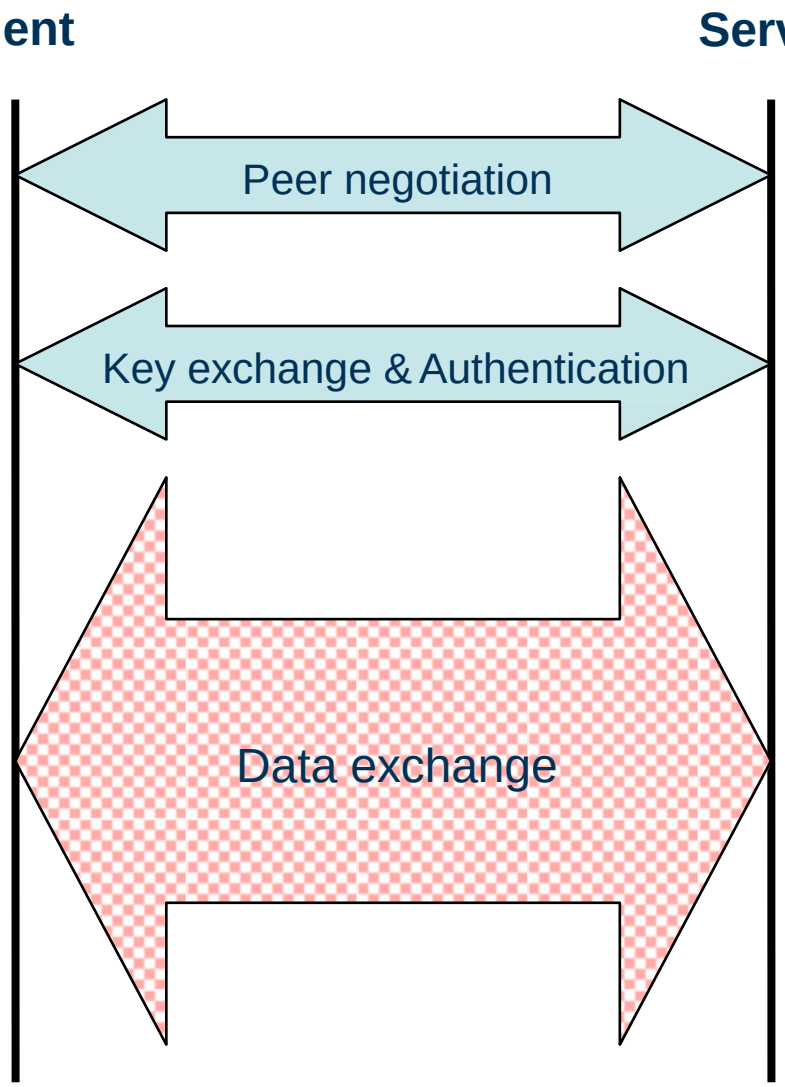
RECORD PROTOCOL

- It takes an application message to be transmitted and then
  - Fragments the data into blocks,
  - Compresses the data (optionally),
  - Calculates a MAC,
  - Encrypts,
  - Adds a header and
  - Transmits it

RECORD PROTOCOL



PROTOCOL PHASES

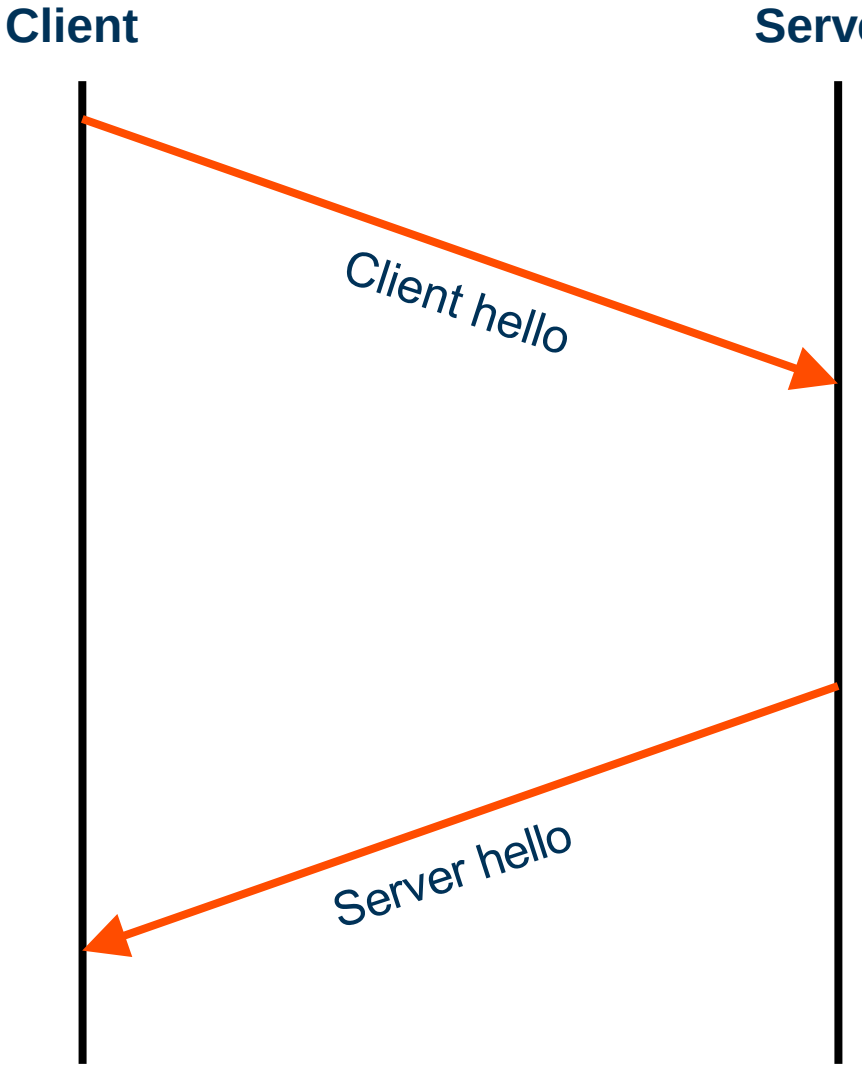


Determine algorithm support

Key exchange
Authentication (certificate-based)
Using public key encryption

Using symmetric encryption

HANDSHAKE PROTOCOL: PEER NEGOTIATION



- SSL version
- Session ID
- Random
- Cipher suite
- Compression method

- SSL version
- Session ID
- Random
- Cipher suite
- Compression method

HANDSHAKE PROTOCOL: PHASE 1

- CLIENT HELLO
  - Random
    - 32-byte value (timestamp)
  - Session ID
    - Non-zero for new connection on existing session
    - Zero for new connection on new session
  - Client version: Highest version
  - Cipher_suite list: Ordered list
  - Compression list: Ordered list

HANDSHAKE PROTOCOL: PHASE 1

- SERVER HELLO
  - Random
    - 32-byte random value
  - Session ID
    - New or reuse
  - Version
    - MIN(client suggested, highest supported)
  - Cipher_suite list: Single choice
  - Compression list: Single choice

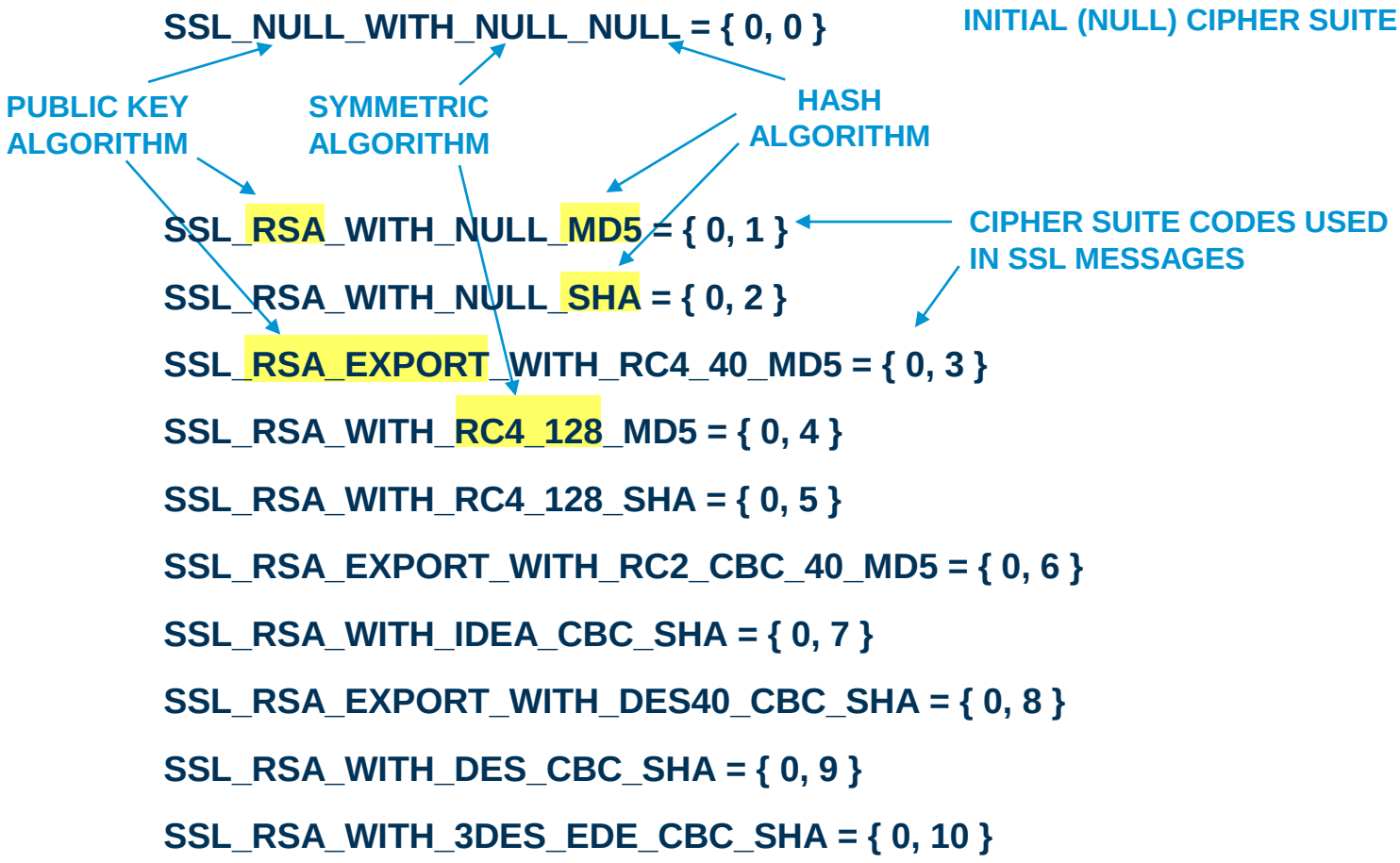
HANDSHAKE PROTOCOL: PHASE 1

- CIPHER SUITE
  - Key exchange method
    - **RSA** requires receiver's public key certificates
    - **Fixed DH** requires both sides to have public key certificates
    - **Ephemeral DH** signed keys are exchanged, need signature keys and public key certificates on both sides
    - **Anonymous DH** requires no authentication of DH keys, susceptible to man-in-the-middle attack
    - **Fortezza** key exchange

HANDSHAKE PROTOCOL: PHASE 1

- CIPHER SUITE
  - Cipher spec
    - CipherAlgorithm: RC4, RC2, DES, 3DES, DES40, IDEA, Fortezza
    - MACAlgorithm: MD5 or SHA-1
    - CipherType: Stream or block
    - HashSize: 16 or 20 bytes
    - IV Size: Size of IV for CBC

CIPHER SUITES: EXAMPLES



TO BE CONTINUED

- See the next lecture

Questions?

Thanks for your attention!