AP/ITEC 2210 3.0 A: System Administration Fall 2023

Instructor: Jamon Camisso
ITEC 2210 Chat: mattermost.itec2210.ca
Email: jamon@yorku.ca
Website: https://eclass.yorku.ca/

Date/Time: Wednesday, 19:00-22:00
Location: Zoom / ACE 003
Office hours: Via Mattermost any time

Class 8 - Encryption continued

- Encryption Review
  - **Symmetric** (AES)
  - **Asymmetric**/Public (RSA, Diffie-Hellman)
  - **Hash Algorithms** (MD5, SHA-{1,2,3})
  - **Signatures -** Private key encrypts a message hash, public key decrypts it, ensuring message authenticity and sender identity, no tampering during transit

Class 8 - Encryption continued

- Encryption Review
  - **Certificate Authorities** (CA) act as a trusted third party
  - CA signs your public key with their private key
  - You can then issue signatures, or receive encrypted messages, for example a TLS webserver
  - If a user has the CA's public certificate, they can reasonably trust your certificate

Class 8 - Encryption continued

- Encryption Review
  - An alternate model to third parties: **Web of Trust**
  - A public key collects signatures from various sources
  - With enough independent signatures you can be reasonably sure the public key belongs to someone
  - However, not used in browsers for TLS (mostly email)

Class 8 - Encryption continued

- RSA - James Grime
- Good review
- Practical sized #s

Class 7 - Encryption, Security

- Now, brief X.509/SSL/TLS certificate aside
- **RSA** is used in this signing and hashing scheme*
- ‘Plaintext’ is set of fields, like domain name, location, expiry dates
- As well as *e* and *n* from the RSA keypair (public key)
- A 3rd party **Certificate Authority (CA)** digitally signs the certificate with their private key, essentially validating you are who you say you are in the certificate

* Elliptic curve and other algorithms can be used, not as widely supported

Class 7 - Encryption, Security

- Now, brief X.509/SSL/TLS certificate aside
- **/etc/ssl/certs** on your VM contains 3rd party CA (certificate authority) certs, tools like curl, apt rely on these for security
- Your browser and OS will also contain a vetted list of CA certificates
- You can validate a certificate that a CA has signed with their private RSA key based on the principle of reversing encryption using the public key
- Say, for example, a bank's TLS certificate, which itself has a public RSA key in it. Use the key to decrypt signatures, and encrypt messages

Class 7 - Encryption, Security

- **Asymmetric encryption**
- Ok, back to Bob and Alice
- What if Alice and Bob have never met, but want to exchange messages using a shared secret key?

Class 8 - Encryption continued

- Asymmetric review

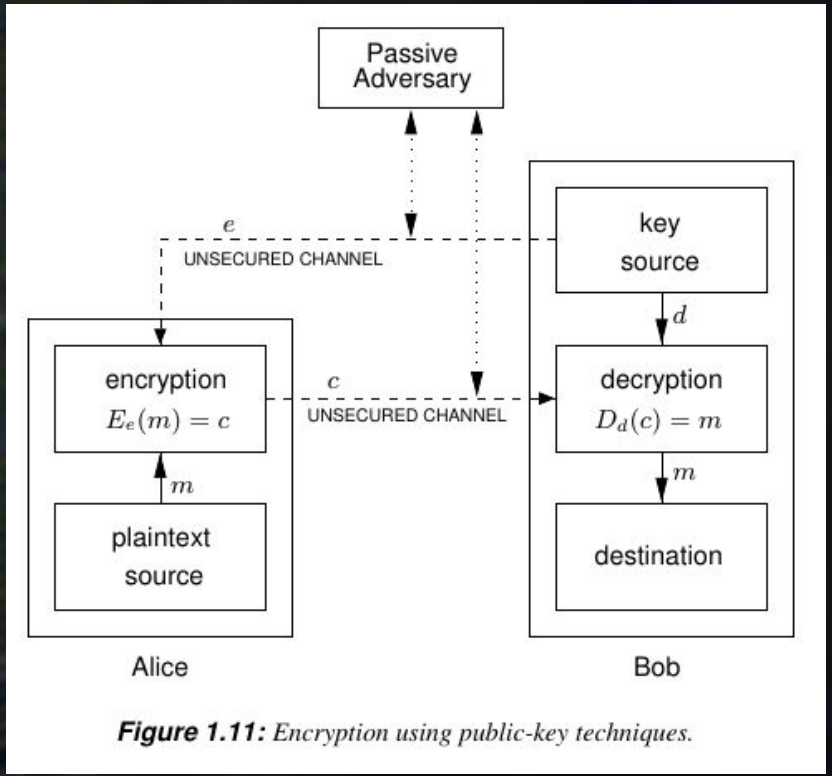
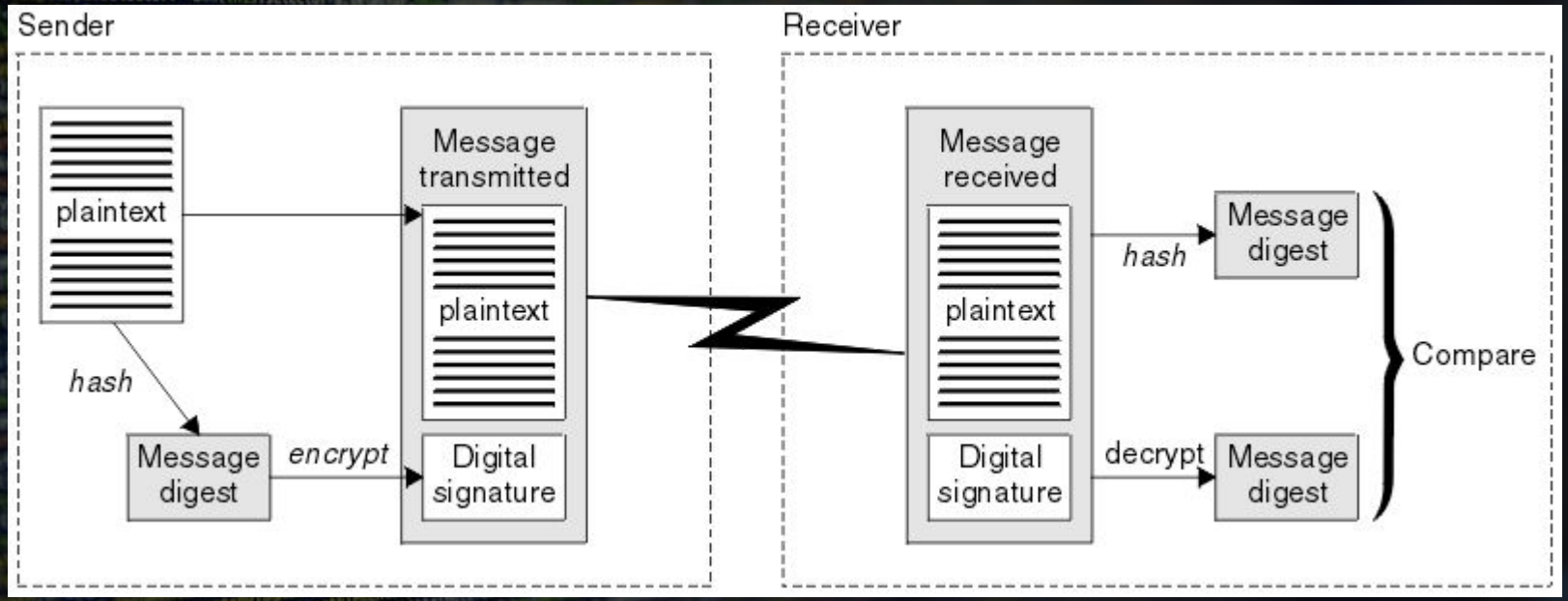


Figure 1.11: *Encryption using public-key techniques.*

Class 8 - Encryption continued

- **Putting it all together**
- What if two parties who have never met would like to exchange secure messages? Say a credit card transaction online
- Recall our tools so far:
  - **Hash functions** - **SHA-1**, **SHA-2**, **SHA-3**, **MD5**
  - **Symmetric ciphers** - block and stream, **AES, ChaCha20**
  - **Asymmetric encryption**
    - **RSA** algorithm
    - **DH** algorithm
  - **X509/SSL/TLS certificates** containing **CA** signed public keys

Class 8 - Encryption continued



Class 8 - Encryption continued

- TLS 1.2 handshake
- **RFC 5246 Section-7.3**

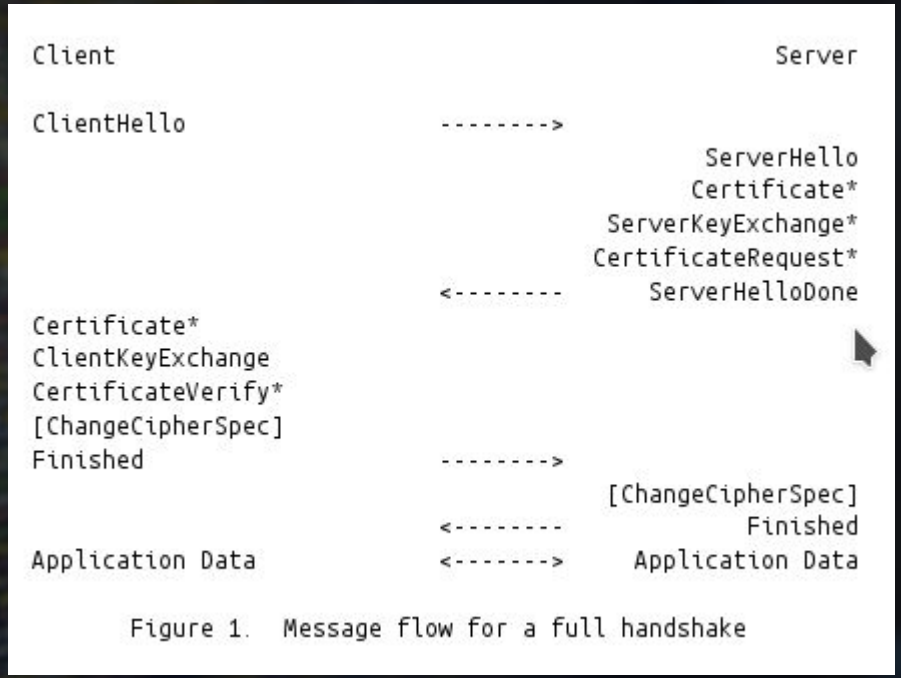


Figure 1. Message flow for a full handshake

Class 8 - Encryption continued

- TLS 1.2 handshake
  - Exchange hello messages to agree on algorithms, exchange random values, and check for session resumption
  - Exchange the necessary cryptographic parameters to allow the client and server to agree on a premaster secret
  - Exchange certificates and cryptographic information to allow the client and server to authenticate themselves

Class 8 - Encryption continued

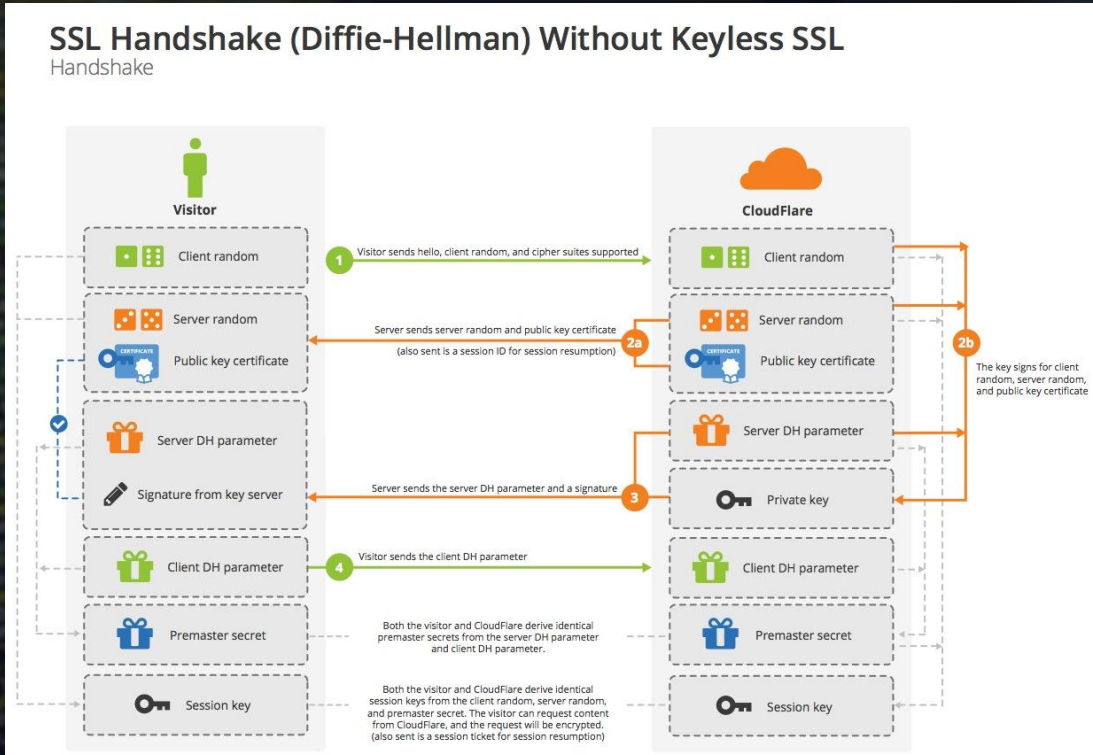
- TLS 1.2 handshake
- Generate a master secret from the premaster secret and exchanged random values
- Provide security parameters to the record layer
- Allow the client and server to verify that their peer has calculated the same security parameters and that the handshake occurred without tampering by an attacker

Class 8 - Encryption continued

- TLS 1.2 handshake
- *Session resumption is useful because the handshake is so heavy (back and forth, cipher selection, random value generation, encryption etc.
- Instead of recalculating every value, if a client has a session ID, the server and client can skip to ChangeCipherSpec and start sending encrypted traffic

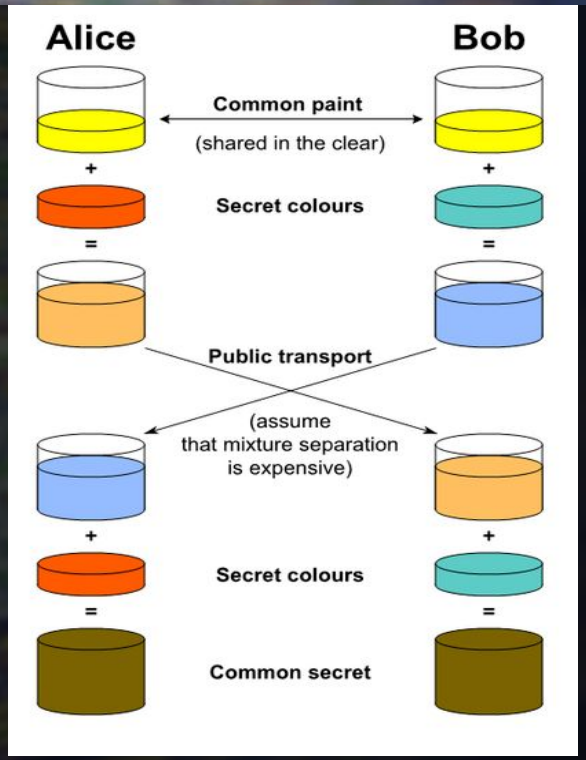
Class 8 - Encryption continued

- TLS 1.2 handshake
- Ignore keyless part
- Note both parties have their own private DH secret



Class 7 - Encryption, Security

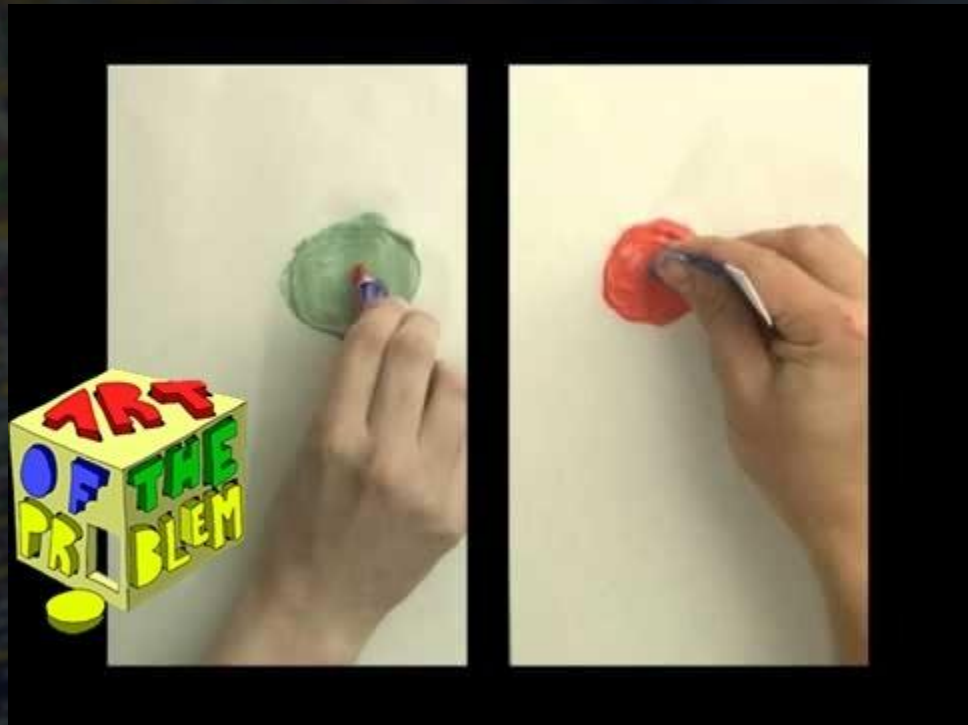
- Enter Diffie-Hellman algorithm



https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

Class 7 - Encryption, Security

- Diffie-Hellman key exchange
- **Logjam vulnerability**
  - **See also cloudflare**



Class 8 - Encryption continued

- TLS 1.2 handshake - **Illustrated Guide**
- For examples see the following wireshark captures:
  - **sudo puppet agent --test --no-daemonize --server instructor.itec2210.ca**
  - **tshark -r /home/itec2210/tls.pcap**
    - Trace through and see each ClientHello with different cipher suites

Class 8 - Encryption continued

- TLS 1.2 handshake
- Note you'll see many ECDHE, DHE, ECDH cipher suites
  - EC means elliptic curve, which is a branch of cryptography that uses curves and functions to select primes
  - Much smaller primes needed when using curves, means faster TLS negotiation, fewer bytes in a handshake

Class 8 - Encryption continued

- TLS 1.2 handshake
- Something like DHE/EDH means Ephemeral Diffie-Hellman
  - No two TLS sessions will share the same key, meaning if a session key is captured, no others are compromised
  - Principle here is called **Perfect Forward Secrecy**
  - https://scotthelme.co.uk/perfect-forward-secrecy/