COSC 301
Network Management and Security

Lecture 16: SSL/TLS

Today's Focus

How to **secure** web applications?

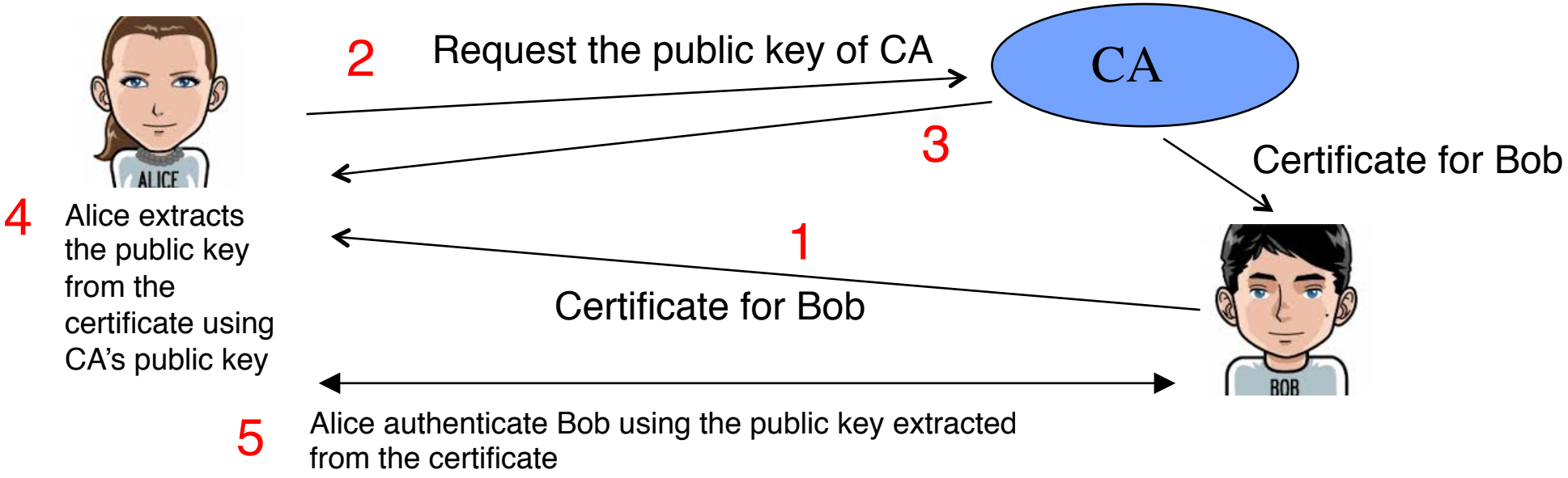
-- Certificate
-- Secure Socket Layer (SSL)
-- Transport Layer Security (TLS)
-- HTTPS

How to get the public key?

- Through certificate issued by CA
- How do we know a website is secure?
  - A closed padlock and https protocol

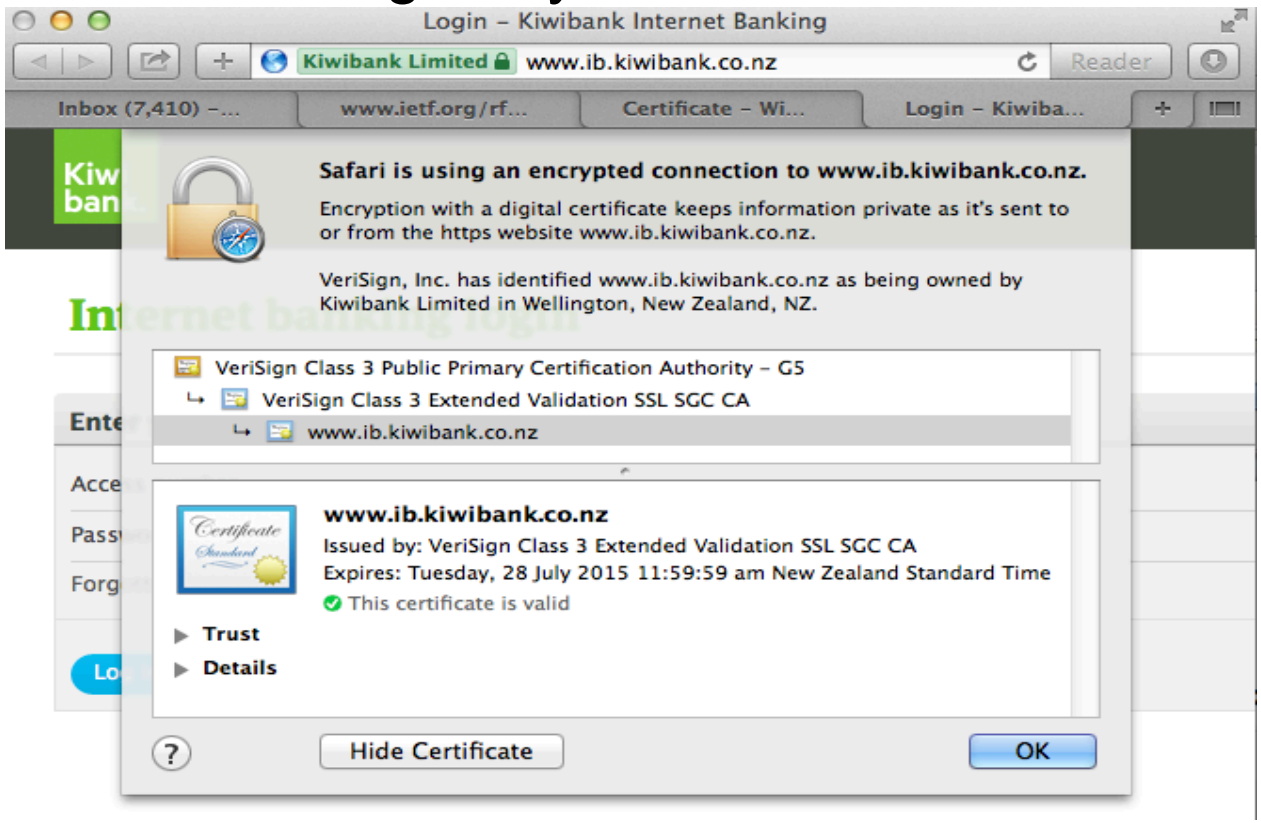
Public Key Distribution

- Certificate Authority (CA)
  - Generate a certificate for Bob with its private key
  - Step 2 is usually not necessary as the public key of the CA is stored locally by the browser as trusted CA



Website Certificate (1)

- What is website certificate?
  - Used to guarantee a website is trustworthy and is the one it claims to be.
  - Generated, signed, and managed by the certificate authorities (CA)

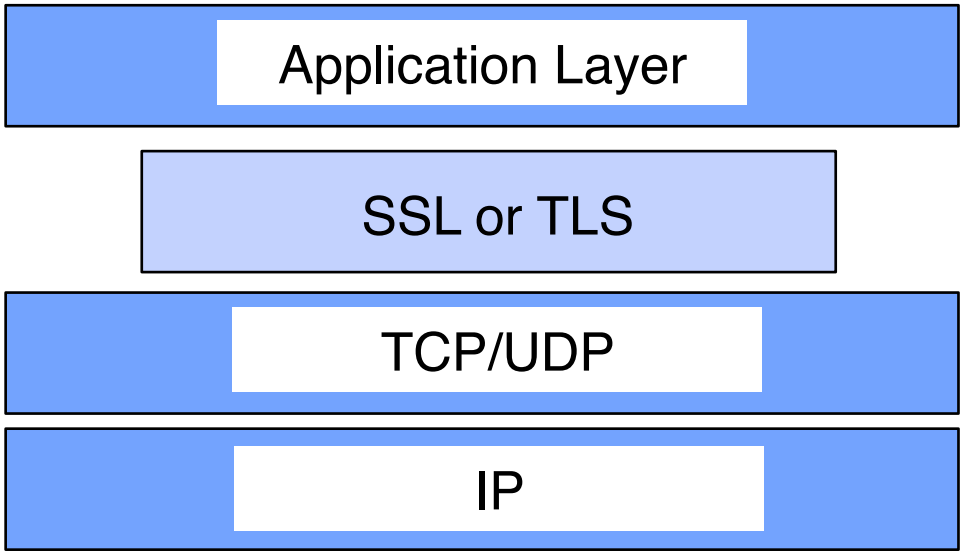


Website Certificate (2)

- Can you trust a certificate?
  - Check if the web address matches the address on the certificate
  - Check if the certificate is signed by a trusted certificate authority, and the date is valid
  - Check the key length, the extensions, the encryption algorithms

SSL and TLS

- SSL – Secure Sockets Layer protocol
- TLS – Transport Layer Security protocol
- To provide security and compression services to data generated by the application layer



Fragmentation
Compression

Message Integrity

Confidentiality

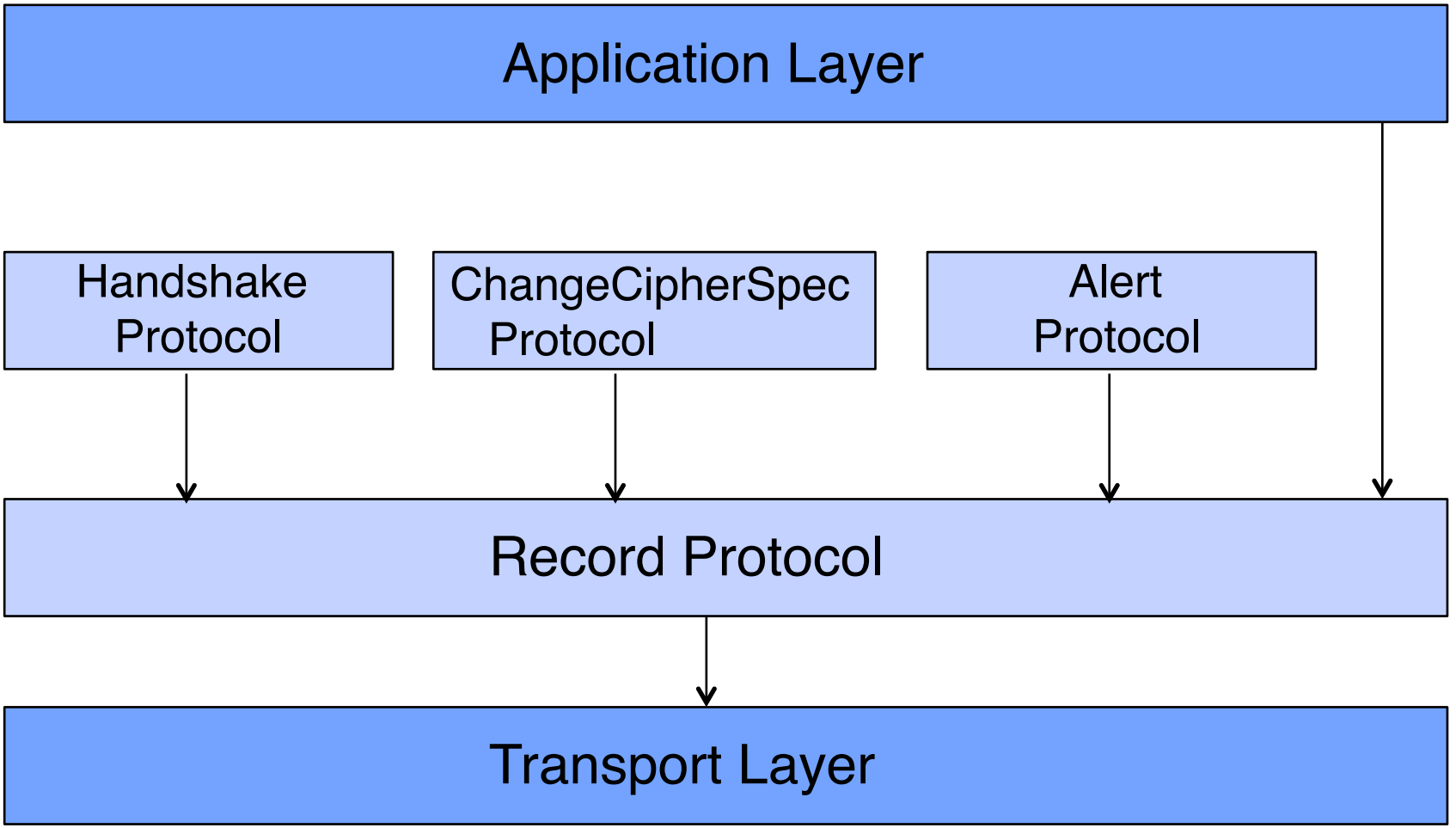
Framing

SSL & TLS History

- SSL v2.0 by Netscape (disable this!)
- SSL v3.0 more scrutiny, fixes attack vectors
- TLS v1.0 (= SSL v3.1) by IETF

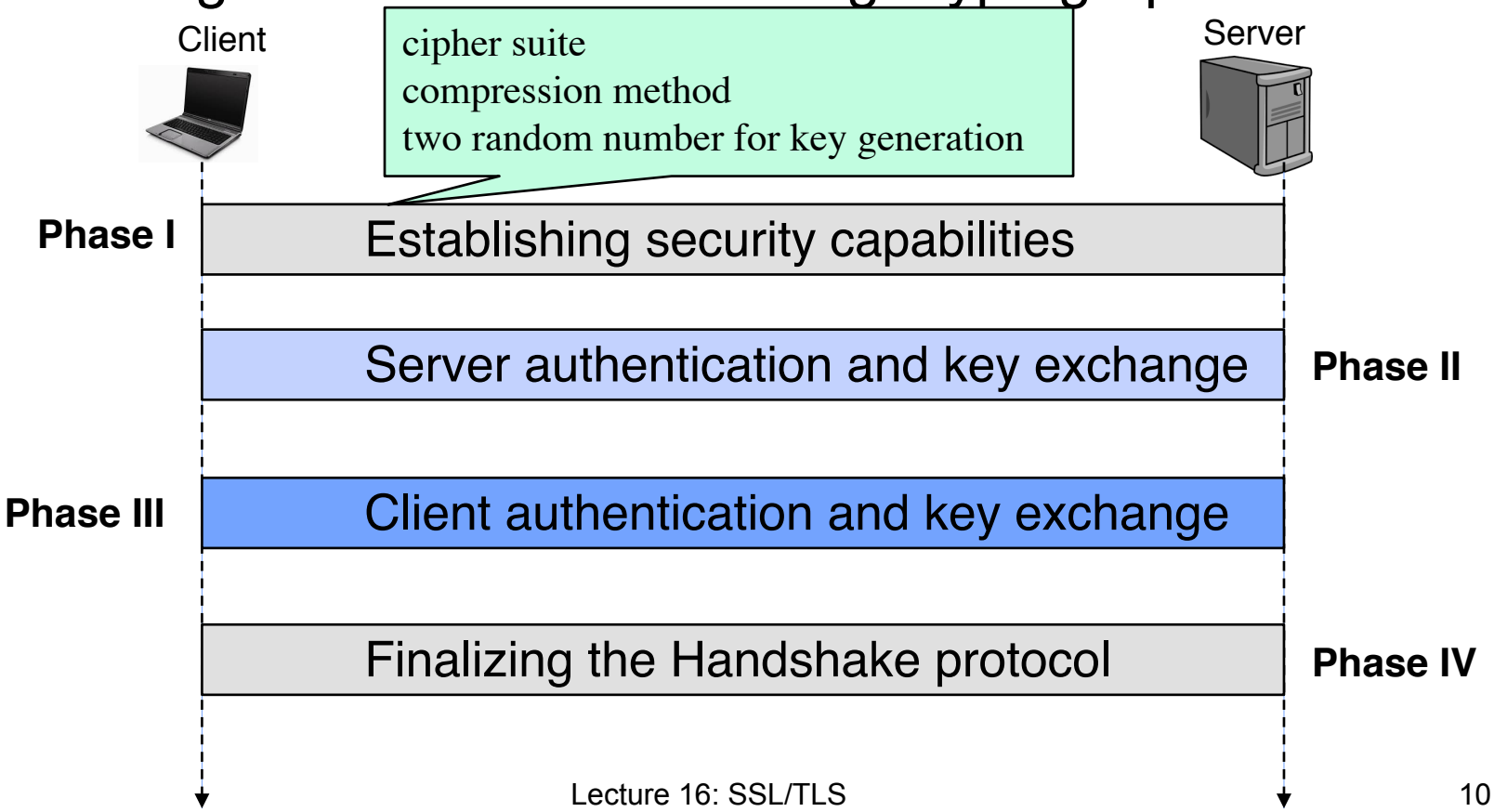
  Few changes, but incompatible with v3.0
- TLS v1.1 clarifies, adds recommendations
- TLS v1.2 cipher updates, extensions
- TLS v1.3 still in draft, delayed due to compatibility issues

SSL/TLS Protocols



The Handshake Protocol

- Uses messages to
  - Negotiate the cipher suite
  - Authenticate sever and/or client with certificate
  - Exchange information for building cryptographic secrets



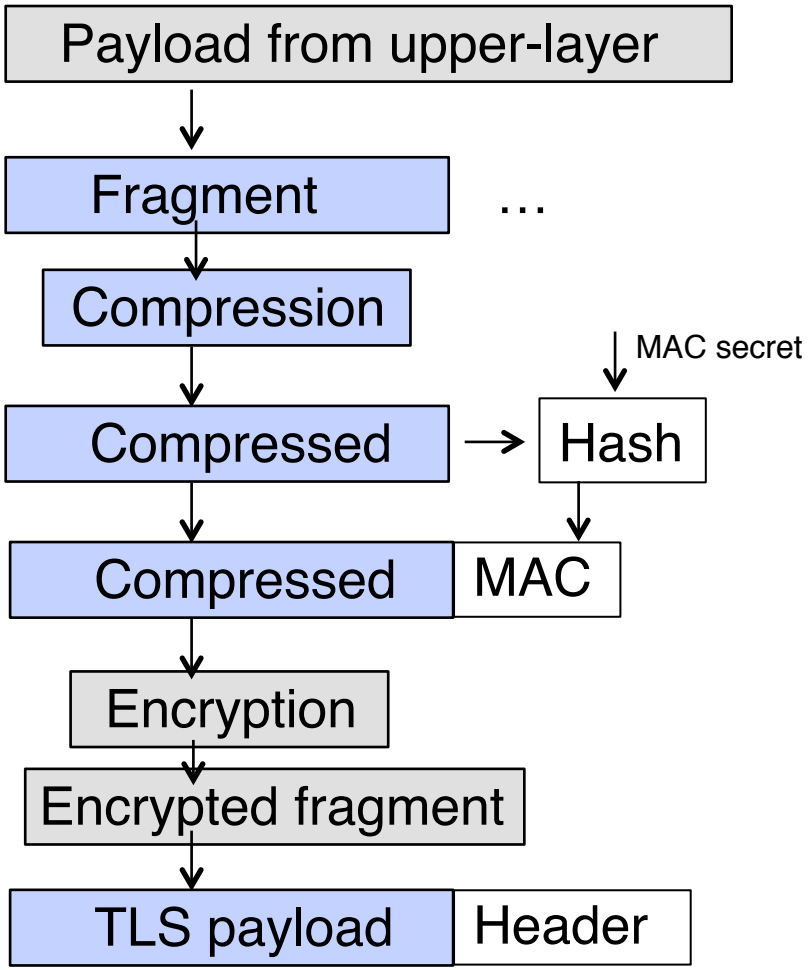
ChangeCipherSpec & Alert Protocols

- When can the two parties use these parameters or secrets?
  - Cannot use them until they have sent or received a special message -> the ChangeCipherSpec message
- How to deal with errors?
  - Uses the Alert protocol to report errors and abnormal conditions.

The Record Protocol

- Carries messages from the upper layers
  - Message fragmentation
  - Message compression (optional)
  - Message encryption

All encrypted except the header!



TLS in action

The client sends a "Client hello" message to the server, along with the client's random value and supported cipher suites.
The server responds by sending a "Server hello" message to the client, along with the server's random value.
The server sends its certificate to the client for authentication and may request a certificate from the client. The server sends the "Server hello done" message.
If the server has requested a certificate from the client, the client sends it.
The client creates a random Pre-Master Secret and encrypts it with the public key from the server's certificate, sending the encrypted Pre-Master Secret to the server.
The server receives the Pre-Master Secret. The server and client each generate the Master Secret and session keys based on the Pre-Master Secret and the random numbers.
The client sends "Change cipher spec" notification to server to indicate that the client will start using the new session keys for hashing and encrypting messages. Client also sends "Client finished" message.
Server receives "Change cipher spec" and switches its record layer security state to symmetric encryption using the session keys. Server sends "Server finished" message to the client.
Client and server can now exchange application data using the Record Protocol over the secured channel they have established. All messages sent from client to server and from server to client are encrypted using session key.

Examples to use TLS

- openssl s_client −starttls smtp −connect smtp.gmail.com:587 −crlf
- openssl s_client −connect smtp.gmail.com:465 −crlf
- Find the public key of a website or its certificate
  - openssl s_client -connect www.otago.ac.nz:443 | openssl x509 -pubkey –noout
  - openssl s_client -connect www.otago.ac.nz:443
- See more details of STARTTLS at https://www.fastmail.com/help/technical/ssltlsstarttls.html

HTTPS

- HTTP over TLS or HTTP over SSL
  - Layering HTTP on top of the SSL or TLS
  - Adding security capabilities of SSL/TLS to standard HTTP

- Difference from HTTP
  - HTTP URLs begin with "http://" and use port 80 by default
  - HTTPS URLs begin with "https://" and use port 443 by default

https://www.ib.kiwibank.co.nz