COMP 3015

Week 8: Public Key Infrastructure (PKI) and Transport Layer Security (TLS)

Week 8 Topics

- Part 1
  - Cryptography Introduction: the art and science of secure communications
    - Symmetric and asymmetric key cryptography
    - Understand the building blocks of cryptography protocols
- Part 2
  - Intro to Transport Layer Security (TLS) – what makes HTTP → HTTPS

Week 8 Topics

This week gets into quite advanced topics with a lot of depth. We’re just aiming for a general, high-level understanding today.

Symmetric Key Cryptography Intro.

The same secret key is used for encryption and decryption for symmetric key ciphers.

In general, this is the interface for symmetric key cryptography functions:

```
let ciphertext = encrypt(plaintext, secretKey)

let plaintext = decrypt(ciphertext, secretKey)
```

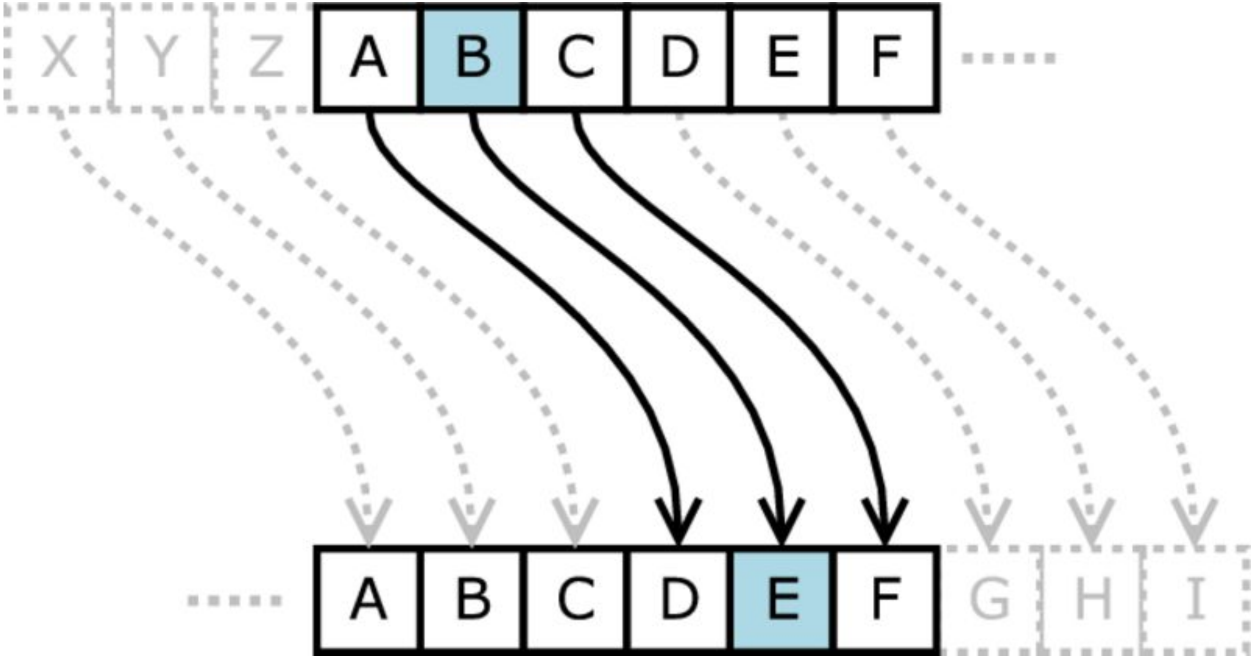
Equivalently:

```
let plaintext = decrypt(encrypt(secretKey, plaintext), secretKey)
```

Famous Cipher: The Caesar Cipher

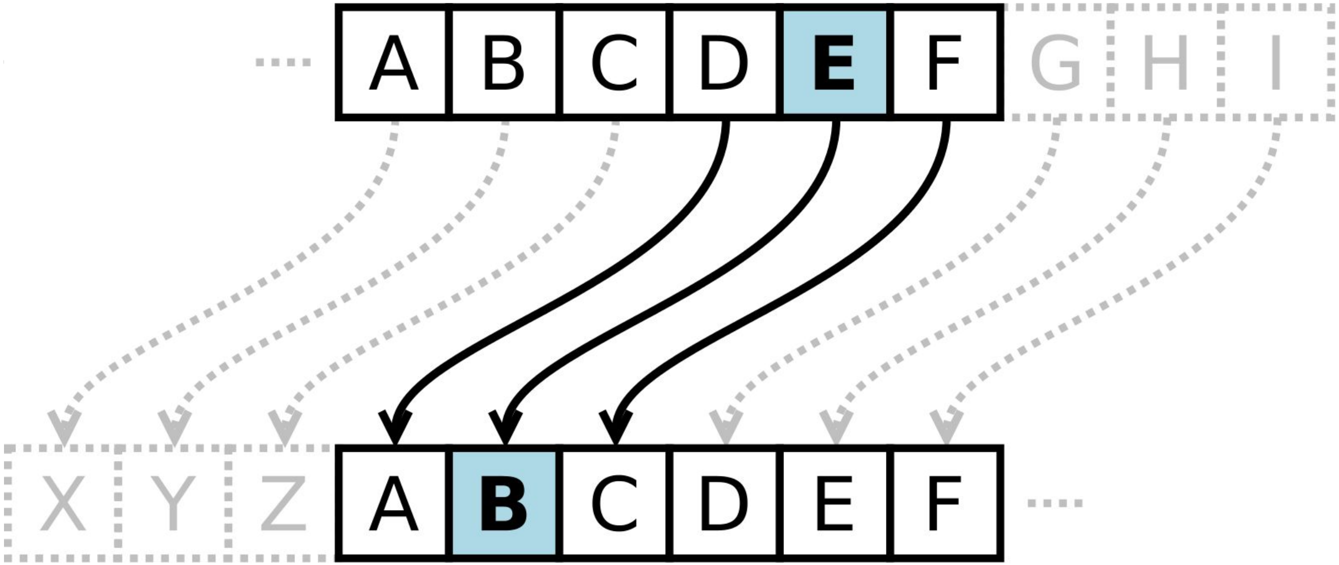
- Named after Julius Caesar who used it to secure communications
- Idea: pick a numeric key such as 3, and use this key to shift plaintext letters to their corresponding ciphertext letters
- Example: plaintext = “ABC”, with a right shift of 3 becomes “DEF”
- Decryption is performed by shifting letters to the left
- “Hello World” of Cryptography

Caesar Cipher Encryption: Shift right by the key



img src: https://commons.wikimedia.org/wiki/File:Caesar_cipher_left_shift_of_3.svg

Caesar Cipher Decryption: Shift left by the key



Caesar Cipher (cont.)

- Today it offers no practical security

- Easy to break the cipher by examining letter frequencies
  - Every language has some letters used much more than others. You can use this info to find the encryption key (the value letters are shifted by)

- Understanding it is the “Hello World” of cryptography

Caesar Cipher (cont.)

See: **CaesarCipher.zip** on D2L

AES Example

Advanced Encryption Standard (AES) is the symmetric key cryptography algorithm that should be used these days.

- Do not write your own algorithms for usage in real systems
  - Use battle-tested industry standard algorithms instead

- We won’t look into the implementation of it since this is not a cryptography course

- See: **AES.zip** on D2L

Asymmetric Key Cryptography Intro.

Idea:

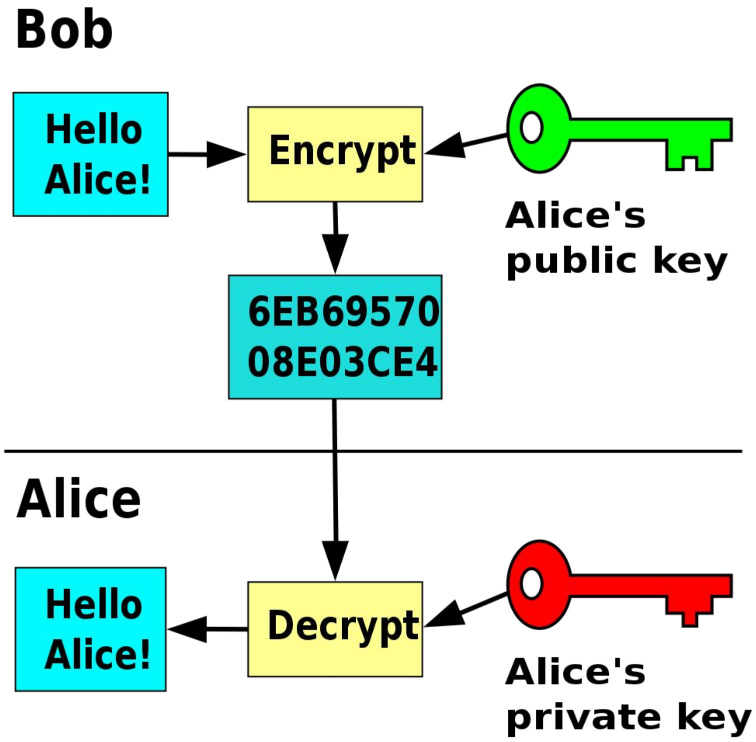
- A keypair is generated
- The keypair consists of a **public key** and a **private key**
- For ensuring confidentiality of data in transit, a message is encrypted using the public key can only be decrypted using the corresponding private key

See: **RSA**

Securing messages in transit with asymmetric key crypto.

After an initial public key exchange (Bob has received Alice's public key, Alice has received Bob's public key):

- Bob writes a message to Alice
- Bob encrypts the message with Alice's public key
- Bob sends the ciphertext to Alice
- Alice uses her private key to decrypt the ciphertext



Before we get into TLS…

Authentication: verify that a user is who they claim to be.

- Example: if you know the username + password to a service, you’re granted access based on the idea that only you should know that combination of data.

Authorization: checks to see if a given user is allowed to perform a certain function or access particular data

- Think clearance levels: classified, secret, top secret, etc.

Transport Layer Security (TLS)

- Cryptographic protocol for secure network communications
- Evolution of Secure Sockets Layer (SSL)
  - TLS version 1 started as a version of SSL
  - You’ll still often see references to SSL, and lots of people use it interchangeably with TLS (incorrectly)
- Can be used to secure email, VoIP, various other forms of messaging

- We are primarily interested in the role it plays in securing HTTP

TLS: Goals

- See: https://www.rfc-editor.org/rfc/rfc8446#section-1

In short:

- **Authentication**: the server side of the channel is always authenticated.
- **Confidentiality**: data sent over the network after connection establishment is only visible to the communicating endpoints.
- **Integrity**: data sent over the channel cannot be modified by an attacker without detection.

HTTPS

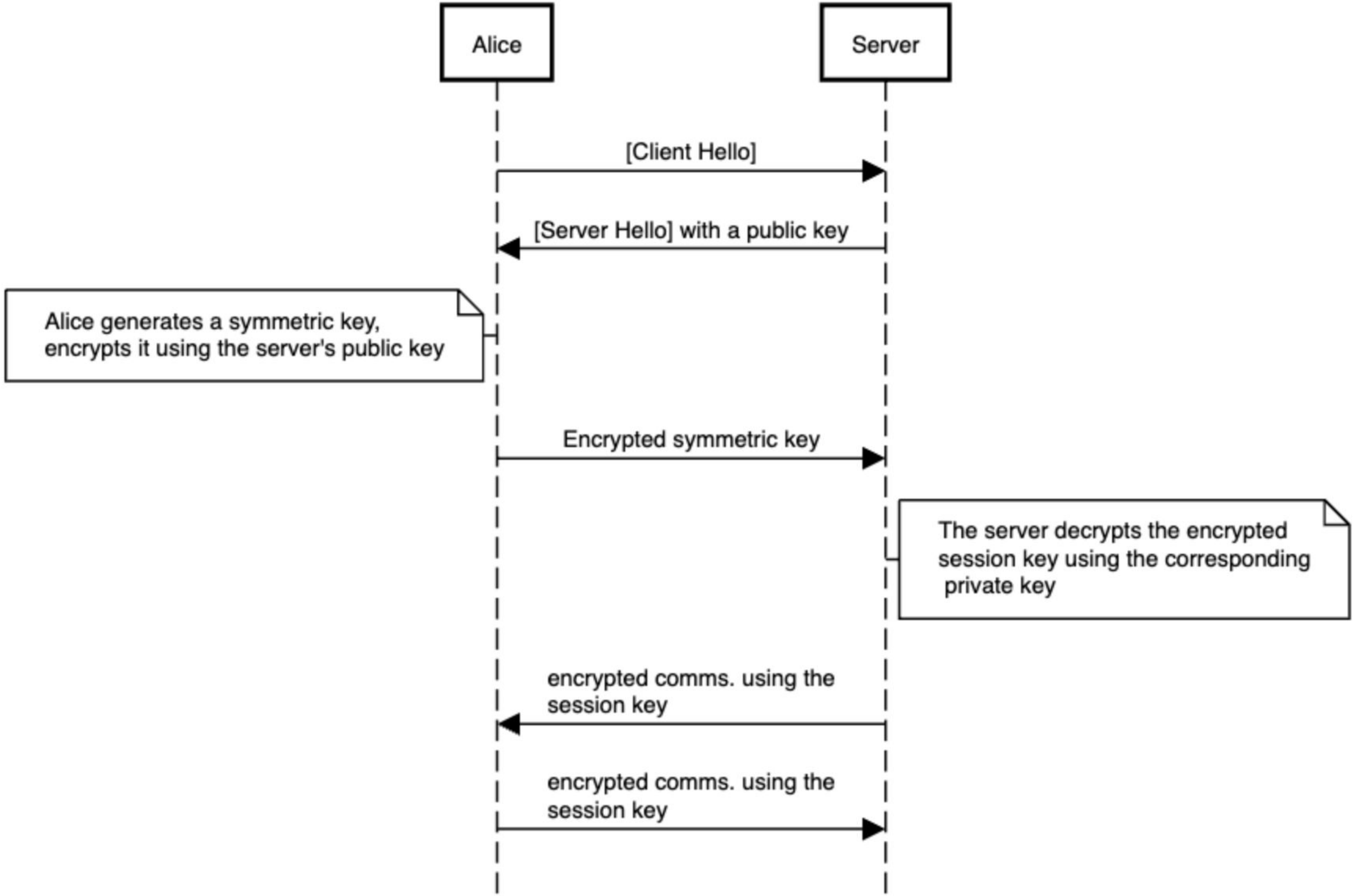
HTTPS is HTTP with TLS

All headers (e.g. Cookie, Set-Cookie), query parameters, route parameters, the body of the HTTP message (e.g. HTTP POST data), etc. is encrypted.

The only plaintext data is the hostname. eg. bcit.ca and lower level protocol information such as the IP addresses, port numbers, etc.

Note: https://blog.cloudflare.com/encrypted-sni

TLS... with a problem. Assume both Alice and the server have generated keypairs



Alice
Server
[Client Hello]
[Server Hello] with a public key

Alice generates a symmetric key, encrypts it using the server's public key

The server decrypts the encrypted session key using the corresponding private key

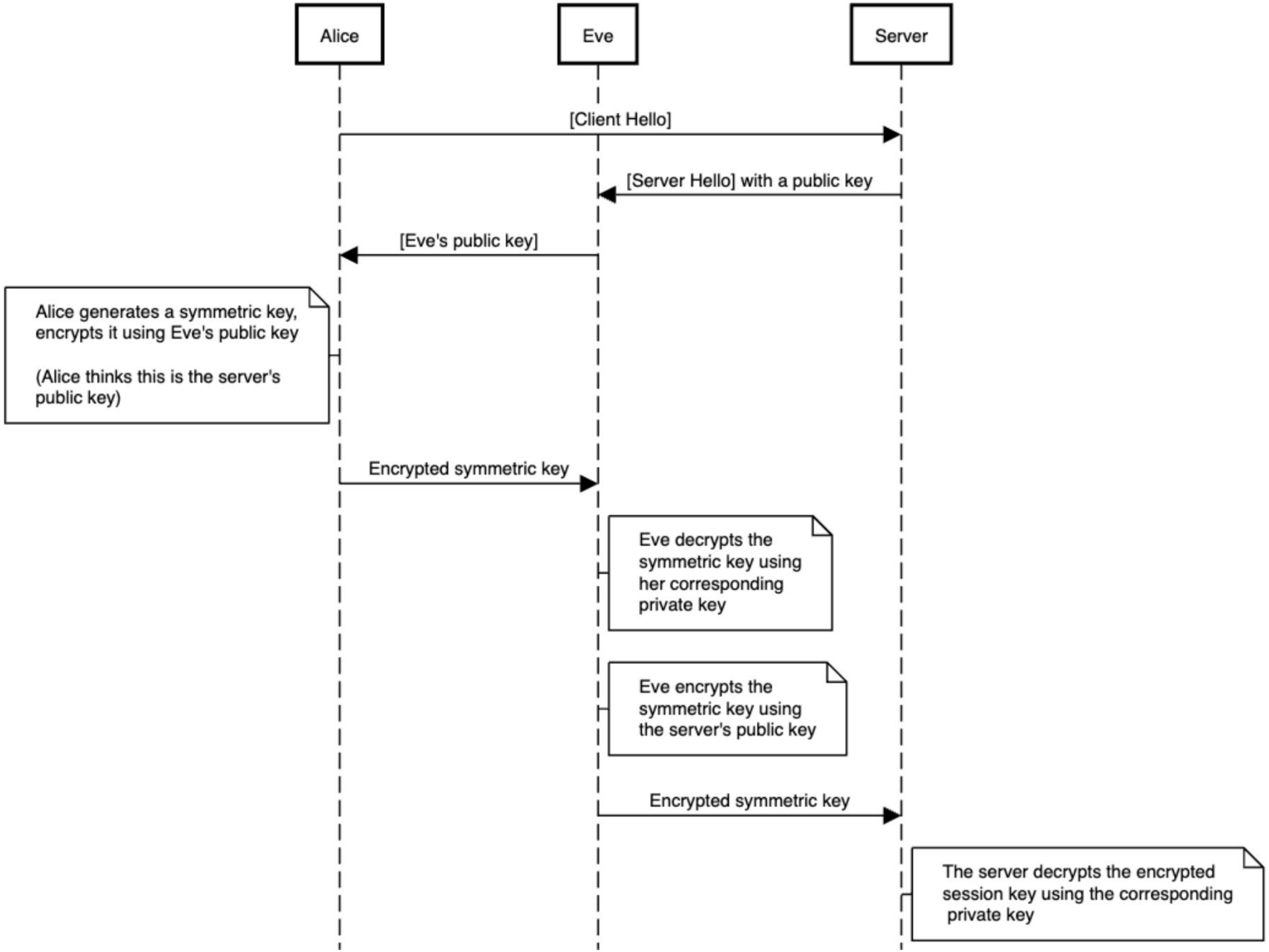
encrypted comms. using the session key
encrypted comms. using the session key

Problem

What about a man-in-the-middle attack?

Traffic is routed through a malicious individual.

Man-in-the-middle attack



Alice

Eve

Server

[Client Hello]

[Server Hello] with a public key

[Eve's public key]

Alice generates a symmetric key, encrypts it using Eve's public key

(Alice thinks this is the server's public key)

Encrypted symmetric key

Eve decrypts the symmetric key using her corresponding private key

Eve encrypts the symmetric key using the server's public key

Encrypted symmetric key

The server decrypts the encrypted session key using the corresponding private key

Key Idea

TLS does **NOT** exchange public keys like shown in the previous slides due to man-in-the-middle (MITM) attacks.

Solution to the MITM attack: The server sends a client a Digital Certificate which contains a public key.

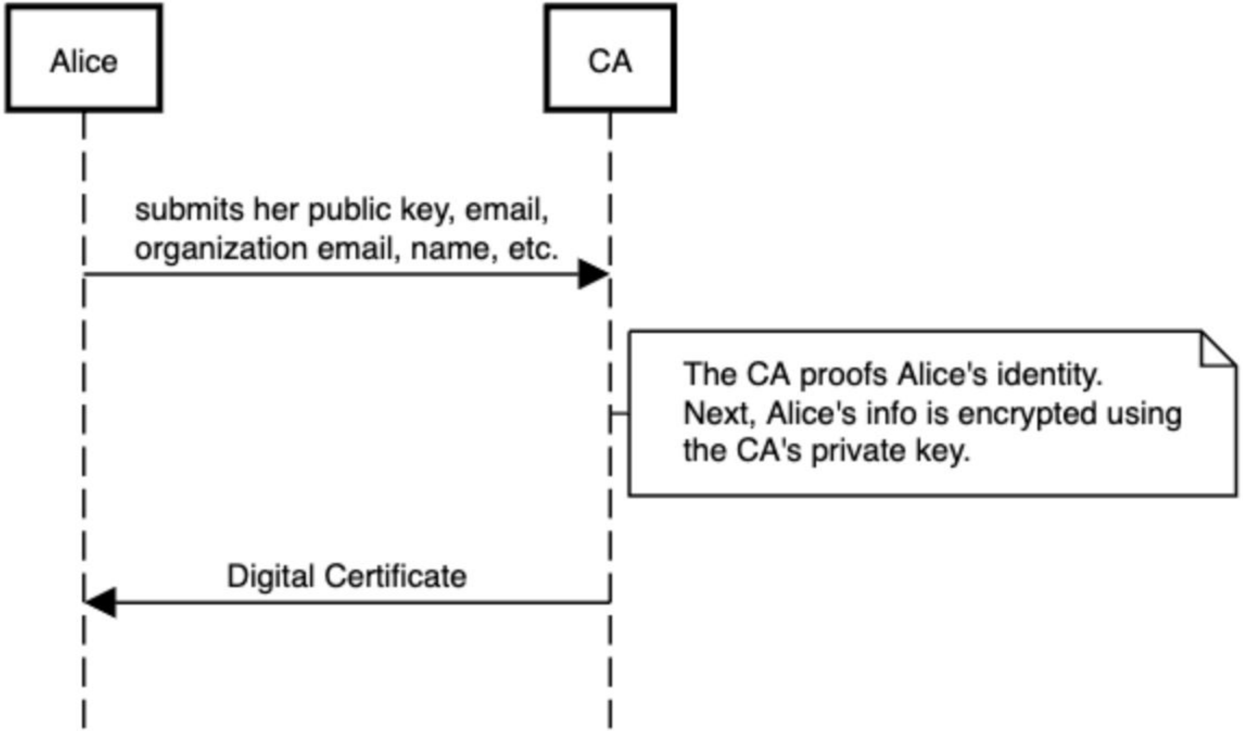
Digital Certificates

- Used to prove/verify ownership of a public key

- Sometimes called “public key certificates”

- Data is encrypted with the private key of a Certificate Authority
    - Everyone can decrypt this since everyone has access to the public keys of the CA (the public keys of the root CAs are loaded onto your computer)

Getting a Digital Certificate



submits her public key, email, organization email, name, etc.

The CA proofs Alice's identity. Next, Alice's info is encrypted using the CA's private key.

Digital Certificate

Getting a Digital Certificate

Note: in order to prove that you own a domain, as part of the verification process a challenge is issued.

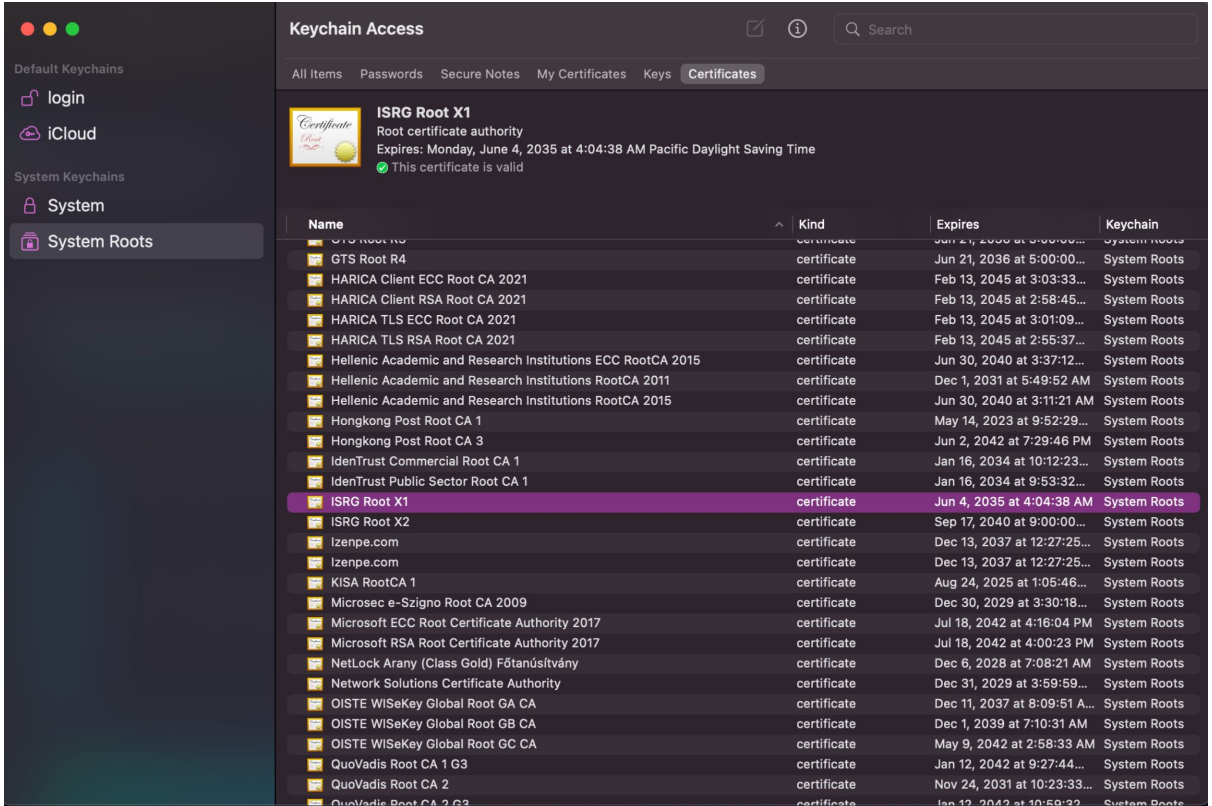
For example: create a DNS record with a given value, or serve a particular HTTP page.

See: https://letsencrypt.org/how-it-works/#domain-validation

Digital Certificates from root CAs

- These provide the public keys of the CA

- The public keys from the CA allow us to verify any certificate which that CA has signed (encrypted using the corresponding private key)

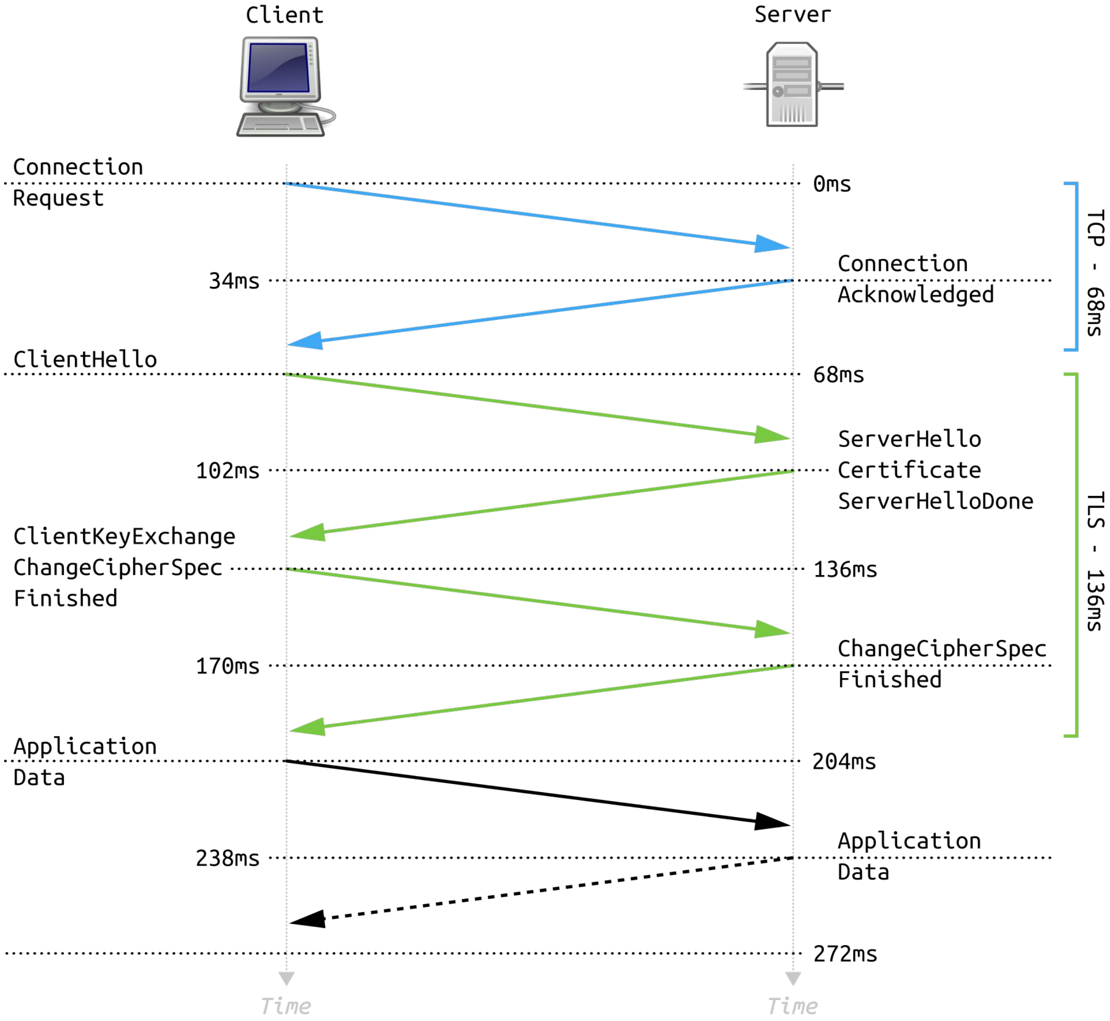


ISRG Root X1
Root certificate authority
Expires: Monday, June 4, 2035 at 4:04:38 AM Pacific Daylight Saving Time
✓ This certificate is valid

Name	Kind	Expires	Keychain
GTS Root R3	certificate	Jun 21, 2036 at 5:00:00...	System Roots
GTS Root R4	certificate	Jun 21, 2036 at 5:00:00...	System Roots
HARICA Client ECC Root CA 2021	certificate	Feb 13, 2045 at 3:03:33...	System Roots
HARICA Client RSA Root CA 2021	certificate	Feb 13, 2045 at 2:58:45...	System Roots
HARICA TLS ECC Root CA 2021	certificate	Feb 13, 2045 at 3:01:09...	System Roots
HARICA TLS RSA Root CA 2021	certificate	Feb 13, 2045 at 2:55:37...	System Roots
Hellenic Academic and Research Institutions ECC RootCA 2015	certificate	Jun 30, 2040 at 3:37:12...	System Roots
Hellenic Academic and Research Institutions RootCA 2011	certificate	Dec 1, 2031 at 5:49:52 AM	System Roots
Hellenic Academic and Research Institutions RootCA 2015	certificate	Jun 30, 2040 at 3:11:21 AM	System Roots
Hongkong Post Root CA 1	certificate	May 14, 2023 at 9:52:29...	System Roots
Hongkong Post Root CA 3	certificate	Jun 2, 2042 at 7:29:46 PM	System Roots
IdenTrust Commercial Root CA 1	certificate	Jan 16, 2034 at 10:12:23...	System Roots
IdenTrust Public Sector Root CA 1	certificate	Jan 16, 2034 at 9:53:32...	System Roots
ISRG Root X1	certificate	Jun 4, 2035 at 4:04:38 AM	System Roots
ISRG Root X2	certificate	Sep 17, 2040 at 9:00:00...	System Roots
Izenpe.com	certificate	Dec 13, 2037 at 12:27:25...	System Roots
Izenpe.com	certificate	Dec 13, 2037 at 12:27:25...	System Roots
KISA RootCA 1	certificate	Aug 24, 2025 at 1:05:46...	System Roots
Microsec e-Szigno Root CA 2009	certificate	Dec 30, 2029 at 3:30:18...	System Roots
Microsoft ECC Root Certificate Authority 2017	certificate	Jul 18, 2042 at 4:16:04 PM	System Roots
Microsoft RSA Root Certificate Authority 2017	certificate	Jul 18, 2042 at 4:00:23 PM	System Roots
NetLock Arany (Class Gold) Főtanúsítvány	certificate	Dec 6, 2028 at 7:08:21 AM	System Roots
Network Solutions Certificate Authority	certificate	Dec 31, 2029 at 3:59:59...	System Roots
OISTE WISeKey Global Root GA CA	certificate	Dec 11, 2037 at 8:09:51 A...	System Roots
OISTE WISeKey Global Root GB CA	certificate	Dec 1, 2039 at 7:10:31 AM	System Roots
OISTE WISeKey Global Root GC CA	certificate	May 9, 2042 at 2:58:33 AM	System Roots
QuoVadis Root CA 1 G3	certificate	Jan 12, 2042 at 9:27:44...	System Roots
QuoVadis Root CA 2	certificate	Nov 24, 2031 at 10:23:33...	System Roots
QuoVadis Root CA 2 G3	certificate	Jan 12, 2042 at 10:59:32	System Roots

TLS 1.2

Released in 2008

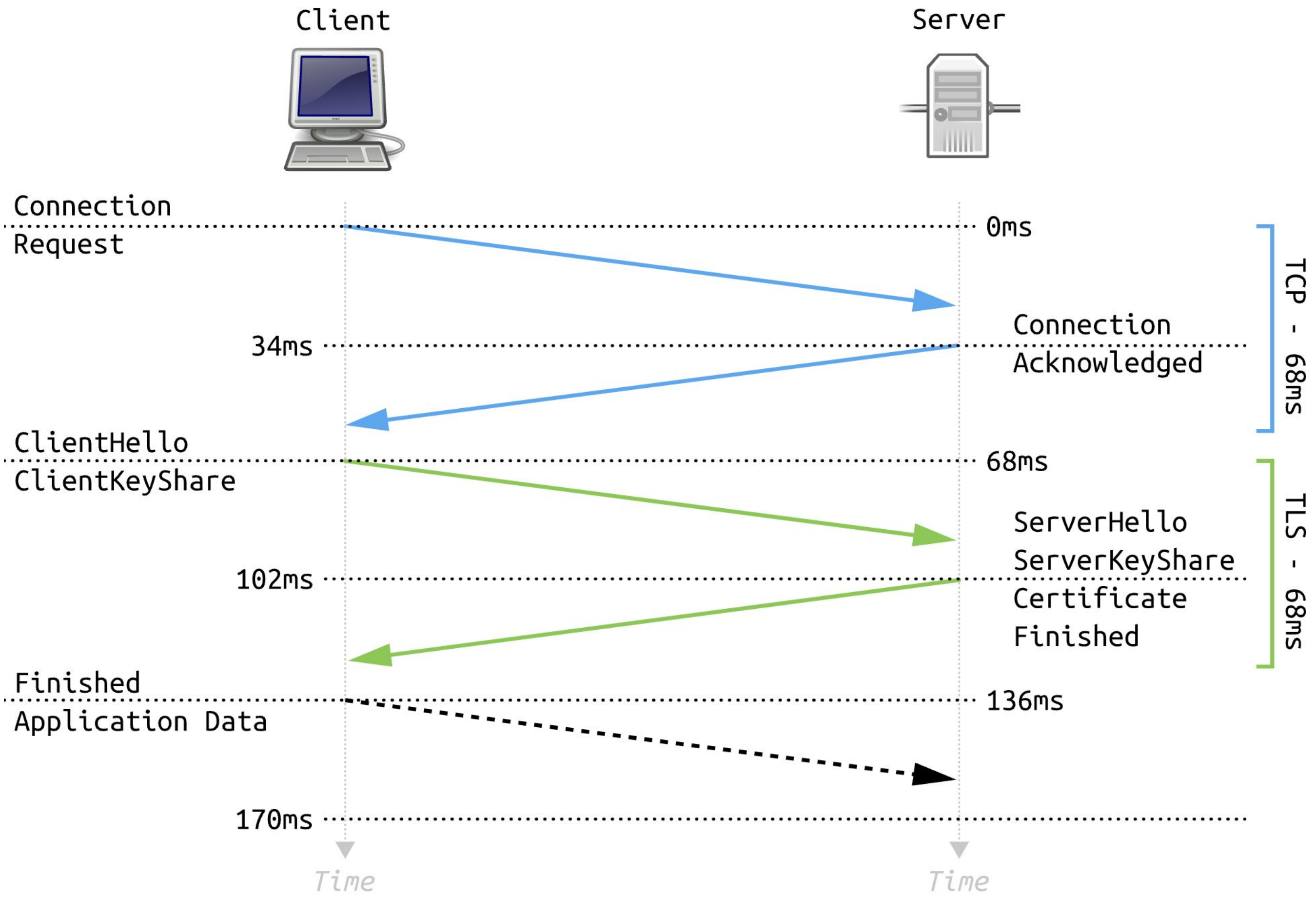


img src: https://commons.wikimedia.org/wiki/File:Full_TLS_1.2_Handshake.svg

TLS 1.3

Released in 2018

One less round trip than TLS 1.2



img src: https://upload.wikimedia.org/wikipedia/commons/7/73/Full_TLS_1.3_Handshake.svg

Certificate Key Replacement?

Can a malicious user replace a legitimate key in a certificate with their own key?

Why or why not?

Certificate Key Replacement?

It’s not possible for a malicious user to simply swap the public key in a digital certificate because it is encrypted using the private key of the CA – remember that the client has the public key, and is therefore able to verify the signature.

Swapping out one public key with another would cause the signature verification to fail.

Let's Encrypt

- https://letsencrypt.org/
- Non-profit Certificate Authority (CA)
- Certificates last 90 days, but you can automatically renew them
  - Use cron to schedule renewals

Quick demo on http://c3015.cfenn.com → let's make it: https://c3015.cfenn.com

Suggested reading: https://letsencrypt.org/how-it-works/

Additional resources on this topic

- https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-22-04
- https://www.cloudflare.com/en-ca/learning/ssl/why-use-tls-1.3/
- In depth look at TLS v1.3: https://blog.cloudflare.com/rfc-8446-aka-tls-1-3

Review

- What is symmetric key cryptography?
- What is asymmetric key cryptography?
- What does TLS stand for?
- What is SSL?
- What HTTP data is encrypted when using HTTPS?
- What is a digital certificate used for?
- When using TLS, what properties does the secure channel provide?

No lab today: focus on assignment #2

Quiz will be posted.

Assignment #3 also is on D2L