Looking up a site

1. DNS lookup: take the URL and find its corresponding IP address
2. TCP handshake
3. Browser sends a formally structured HTTP GET request
4. Browser checks for certificate and validates it (using TLS, only on https)
5. Check the browser cache to see if we've been here before
6. Server sends the HTML of the homepage to the browser
7. Later queries for images, CSS files, JS files, etc

IPs

- → Internet Protocol (IP) gets data from a source machine to a destination machine
- → each piece of communication is stored in an IP packet
- → the IP header contains the source IP address, destination IP address, and other info
- → each computer connected to the network has one or more IP addresses, or unique strings of characters that identify a computer
- → IPv4 uses a 32-bit address, while IPv6 uses a 128-bit address, so it offers 10^{28} times more addresses (don't need to worry about running out)

TCP Handshake

- → Transmission Control Protocol (TCP) is a standard for delivering data through networks that gets data from a source program to a destination program and guarantees in-order, no corruption delivery
- → different ports are used to identify which server the client wants to talk to
- → the TCP header includes the source port, the destination port, the sequence number (how much data is sent), and other fields
- → the TCP Handshake sets up a duplex (two-way) connection between the client software and the server software
  - ↳ the client sends a synchronize Sequence Number (SYN) to the server to let it know that it wants to establish a connection and what sequence number it starts segments with
  - ↳ the server sends back a SYN-ACK, where ACK is the acknowledgement that it received the client's message and SYN signifies the server's sequence number
  - ↳ the client acknowledges the response of the server with ACK and they both establish a reliable connection
- → allows both ends to negotiate the parameters of the socket connection and agree on the initial sequence numbers

HTTP and HTTPS

- → Hypertext transfer protocol (HTTP) is a syntax used to perform request and response between the client software and the server software

target endpoint → `/hello/`; headers → User-Agent, Host, Accept-Language; status message → `200 OK`, `404 Not Found`

```
GET /hello/ HTTP/1.1
User-Agent: Mozilla...
Host: www.carleton.edu
Accept-Language: en
```
HTTP request

```
HTTP/1.1 200 OK
Date: Mon, 9 Oct 2023
Server: Apache...
Content-Length: 88
```
HTTP response

```
HTTP/1.1 404 Not Found
Date: Mon, 9 Oct 2023
Server: Apache...
Content-Length: 230
```
HTTP response

- → Hypertext Transfer Protocol Secure (HTTPS) is an extension of HTTP where communications are encrypted by SSL/TLS

SSL and TLS

- → Secure Socket Layer (SSL) is an encryption-based internet security protocol to establish secure links between networked computers
- → Transport Layer Security (TLS) fixed the vulnerabilities in SSL
- → TLS authenticates more efficiently than SSL
- → The purpose of TLS is to get the client and the server talking, while encrypting the communication so third parties can't eavesdrop
- → during a TLS handshake, the client and the server specify which version of TLS and which cipher suites they will use, authenticate the identity of the server via the server's public key and the SSL CA's digital signature, and generate session keys in order to use symmetric encryption

base64

- → base64 is a method of representing binary data in sequences of 24 bits (3 bytes) that are represented by four 6-bit digits that are ASCII characters

  1011000|01 1100|0110 00|111010| 011100|1000000
  - ↳ S C Y 6 c g
  - ↳ pad so that length is a multiple of 4: SCY6cg==
- → used often on the web to embed image files inside HTML/CSS files
- → essentially no utility software modifies bytes representing printable ASCII chars, which is why base64 is helpful, because data will not be corrupted

Symmetric encryption

- → In symmetric encryption algorithms, Alice and Bob agree on an algorithm E and a key K, and use the same algorithm and key for both encryption and decryption
  - ↳ for plaintext P, $D_K(E_K(P)) = P$, or ciphertext $C = E_K(P)$ and $D_K(C) = P$
- → if P is too long to fit into the input of E_K, then you have to break P into $P_1, P_2, \dots, P_N$ and encrypt those pieces separately
- → a stream cipher generates a stream of random Bs so that $P_i \otimes B$ looks random but P_i is recoverable by doing $(P_i \otimes B) \otimes B$
  - ↳ very fast to compute
  - ↳ examples of stream ciphers are RC4 (now broken), ChaCha
- → a block cipher breaks the plaintext P into blocks of size b (in bits) and uses the bits of the key to decide how to reversibly scramble the bits of P
  - ↳ each operation on a P_i keeps it at b bits, is invertible, and depends on K
  - ↳ examples of block ciphers include AES, DES, TwoFish, TripleDES
- → some modes of block ciphers include
  - ↳ Electronic Code Book (ECB): do not use, just encrypts each block separately which opens it up to frequency attacks due to repetition
  - ↳ Cipher Block Chaining (CBC) takes a key K and an initialization vector IV and encrypts each block using the previous blocks' encryptions: $C_1 = E_K(P_1 \otimes IV)$, $C_2 = E_K(P_2 \otimes C_1)$, $C_3 = E_K(P_3 \otimes C_2)$, ... which means that two identical blocks in the message will get encrypted differently in a hard-to-predict way
- → In 1997, NIST announced a competition to find a successor to DES to be known as AES, asking the public to come up with a block cipher supporting a block size of 128 bits and key sizes of 128, 192, and 256 bits, Rijndael was chosen as the AES

Diffie-Hellman

- → key exchange problem: how do Alice and Bob agree on a key for AES and make sure that no one else has it?

Alice	Eve	Bob
g, p	→ g, p →	g, p
random a		random b
$A = g^a \bmod p$	← B ←	$B = g^b \bmod p$
$K = B^a \bmod p$	→ A →	$K = A^b \bmod p$

- → unless eve can determine a or b, she cannot find K and it is not easy to brute force
- → however, this does not prevent person in the middle attacks

Alice		Mal		Bob
g_1, p_1	→ g_1, p_1 →	g_1, p_1, g_2, p_2	→ g_2, p_2 →	g_2, p_2
$A = g_1^a \bmod p_1$	← M_1 ←	$M_1 = g_1^m \bmod p_1$, $M_2 = g_2^m \bmod p_2$	← B ←	$B = g_2^b \bmod p_2$
$K_1 = M_1^a \bmod p_1$	→ A →	$K_1 = A^m \bmod p_1$, $K_2 = B^m \bmod p_2$	→ M_2 →	$K_2 = M_2^b \bmod p_2$
no idea				no idea

$K_1 \neq K_2$ because likely $a \neq m \neq b$

- → DH is good because neither can unilaterally control the secret key

RSA and Asymmetric Encryption

- → each communicator creates two keys: a secret key S and a public key P
- → given an agreed upon encryption function E, S and P have the properties $E(S, E(P, M)) = M$ and $E(P, (E, S, M)) = M$
- → Alice encrypts her message M with Bob's public key: $C = E(P_B, M)$ and Bob decrypts with his secret key: $M = E(S_B, C)$
- → RSA involves picking two really large prime numbers p and q, let $n = p \cdot q$, then find $e < (p-1)(q-1)$ such that $GCD(e, (p-1)(q-1))$ is 1, and find d such that $ed = 1 \bmod ((p-1)(q-1))$. Then, $S = (n, d)$ and $P = (n, e)$
  - ↳ there is no known way to efficiently compute d from (n, e) so as long as Bob keeps S_B hidden, only Bob can decrypt C
- → If Bob publishes his public key, then anyone can send him a message, but he has no way of guaranteeing they are who they say they are
  - ↳ unless Alice sends $C = E(P_B, E(S_A, M))$, since Bob can decrypt w/ S_B and P_A, and only Alice could have written since only she has S_A
- → this means anyone can write confidentially to anyone else w/o first agreeing on K
- → provides a free digital signature, as long as you trust that the publisher of the public key is who they say they are
- → caveat: the length of M has to be less than n
- → RSA is only able to encrypt data ≤ the size of the key (2048 bits = 256 bytes)

RSA vs AES

- → RSA needs only N keys for N people, AES needs N(N-1)
- → AES has same security w/ shorter keys, making it faster to encrypt/decrypt
- → RSA depends on one-way trapdoor functions that are easy to compute one way but hard the other unless you can factor n quickly
- → Typical scheme is a hybrid: use RSA to exchange a shared key, then use that key with AES to really talk
- → Public key encryption is exclusively used for shorter messages, like hash digests

Cryptographic Hashes

- → cryptographic hash functions take in any # of bytes of input data and use some hash function to map it to a fixed length digest
- → need to be deterministic (same data → same digest), fast, collision resistant, input sensitive (small changes to input yield large changes to digest), and pre-image resistant (hard to reverse engineer to get M from D)
- → the main purpose is to check data integrity, like when downloading Kali
  - ↳ Alice sends M || H(M) so Bob can make sure nothing was corrupted
- → can also be used for digital signatures
  - ↳ Alice sends $M \,||\, E(S_A, H(M))$ so Bob knows nothing was changed and the person with Alice's secret key is the only one who sent it
- → other applications are password storage and proof of work (bitcoin)
- → the function SHA-256 breaks input data into chunks, and for each chunk there is some bit manipulation, and these results are added to the hash value
- → other examples are SHA-2, SHA-1, MD5 (obsolete), BLAKE
- → Message Authentication Codes (MAC) are hashes with a secret key so that you can encrypt the data and hash so that Mal can't just throw away the original M || H(M) with her own message and hash
  - ↳ Alice sends M || MAC(K, M) to Bob, Mal doesn't have K

Certificate Authorities

- → a trusted third party that validates that Bob's public key is actually owned by the real person Bob, rather than Mal, and provides him a certificate

Certificate	
Subject: bob.com Public key: P_B ...	} TBS ("to be signed")
$E(S_{CA}, H(TBS))$	} sig

- → everyone has E, H, and TBS and hopefully only the CA has S_{CA}
- → Alice can validate the certificate by making sure $H(TBS) = E(P_{CA}, sig)$
- → Now Alice can be confident that P_B actually belongs to Bob!
- → typically an asymmetry between Alice (browser) and Bob (web server) so Alice doesn't necessarily need a certificate for her keys

Representing Keys in Files

- → asymmetric encryption key files require the ability to store integers of arbitrary length and represent keys in a bit for bit reliable way
- → Abstract Syntax Notation (ASN.1) is a way to formally describe the file format
- → Then Distinguished Encoding Rules (DER) are used to encode an instance as an unambiguous byte sequence; result is non-ASCII
- → The file format Privacy-Enhanced Mail (PEM) packages this as ASCII for ease of visual examination and safety of transfer (avoid corruption)
- → the pem file includes a descriptive header ("---begin key---"), then includes the key in base64, then ends with a descriptive footer ("---end key---")
- → DER encoding would represent 0 as 02 01 00, where
  - 02 ↳ type = integer
  - 01 ↳ length = 1 byte
    - 82 in this position means 1000 0010 (↳ long, ↳ two octets of length)
  - 00 ↳ value = 0

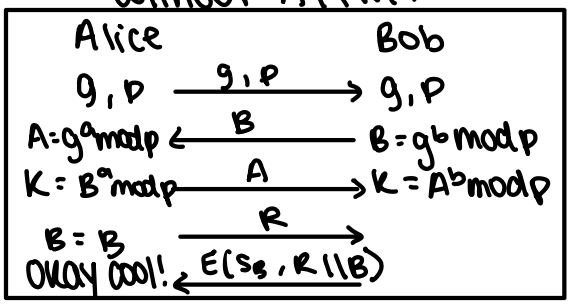
Miscellaneous

→ the git status command displays the state of the working directory and the staging area; so which changes have been staged and which haven't
→ git add: changes to staging area; git commit: permanent screenshot, git push: staging area to remote repo, git pull: remote repo to working directory
→ wireshark is a protocol analyzer/packet sniffer tool that enables us to watch data transmitted to and from our computer
→ kali Linux is an open source penetration testing platform
→ protocol standards are not given the okay until multiple interoperable clients and servers are produced from them, to make sure they are specific enough
→ the CIA triad is confidentiality (M can't be read by attackers), integrity (M can't be modified), and authenticity (M came from the claimed party)
→ authentication is who are you, and prove it, while authorization is now that I know you, what am I able to show you
→ both the software and the machine it is running on are known as the server

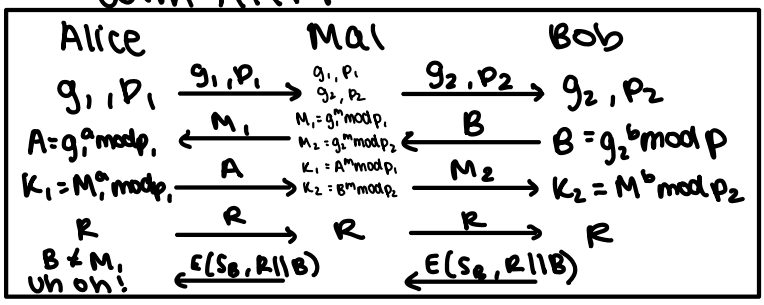
Preventing AITM

→ After doing DH, Alice will send Bob back a random R to encrypt to make sure that there is no attacker in the middle

Without AITM:



With AITM:



→ Mal can't change Bob's response because she doesn't have S_B, so she is bound to be foiled when Alice realizes Bob has a different B than she does
→ realistically, the response will be $E(S_B, H(R||B))$ because of RSA's size limit