Applied Cryptography and Web Security

Furkan Alaca
furkan.alaca@carleton.ca

Carleton University

SYSC 4700, Winter 2018

Learning Objectives

- What are the goals of computer/network security?
- Review: Symmetric- and public-key encryption
- PKI, secure web browsing (TLS/HTTPS)
- Challenges facing secure online communication
  - Case studies: secure web browsing, secure messaging

Computer Security Goals

Computer security is based on the following principles:

- Confidentiality: Ensures that information is only accessible to authorized parties
- Integrity: Ensures that information has not been tampered with
  - Also that network services operate as intended
- Authenticity: Ensures that the originator of a received message is known
- Non-repudiation: Ensures that no entity can credibly deny having sent a message or performing an action
- Availability: Ensures the availability of network communication and network services to all authorized parties

Review: Cryptographic Tools

- Symmetric encryption:
  - Same key used for encryption and decryption
  - Sender and receiver must securely establish a shared secret key
  - Can be a **block cipher** or a **stream cipher**
  - Guarantees **confidentiality**
- Public-key encryption:
  - One key is used for encryption, the other for decryption
  - If Alice encrypts a message with Bob's public key, **only** Bob can decrypt the message, using his private key
  - Guarantees **confidentiality**
- Digital signatures:
  - Alice **signs** a message using her private key, and sends it to Bob
    - The message itself may or may not be encrypted
  - Bob **verifies** the signature using Alice's public key
  - Guarantees **source authenticity**, **integrity**, and **non-repudiation**

Review: Cryptographic Tools (2)

- Cryptographic hash functions:
  - A hash function maps any variable-length input to a fixed-length output
  - A **cryptographic** hash function must be:
    - Computationally efficient in computing the hash of a message
    - **Infeasible** to recover the original message from its hash
    - Given the hash of a message, it is **infeasible** to find another message with the same hash
    - It is **infeasible** to find two different messages with the same hash
- Message Authentication Codes (MAC):
  - Computed and verified using a **shared** secret key
  - Guarantees the **integrity** and **source authenticity** of a message
  - Often implemented by embedding the key into the message and then computing the hash (known as HMAC)
- Authenticated Encryption:
  - A block cipher **mode of operation** which provides both (1) the confidentiality guarantees of symmetric encryption and (2) the integrity and authenticity guarantees of MACs

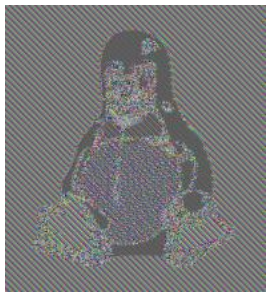
Security Properties of Encryption Algorithms

- For symmetric encryption:
  - Should be secure against **cryptanalysis**, i.e., the attacker should be unable to decrypt ciphertext or discover the key, even when in possession of a large collection of ciphertext-plaintext mappings
  - Key size should be at least 128 bits, to resist **brute-force attacks** which iterate through all possible keys
- For public-key encryption:
  - Resistant towards cryptanalysis
  - Infeasible to compute the private key, using knowledge of the public key
  - Key size should be 2048 bits for most public-key encryption (elliptic curve cryptography requires less)
- Popular algorithms which satisfy modern security requirements:
  - Public-key encryption: RSA
  - Digital signatures: DSA, ECDSA
  - Symmetric-key encryption: AES (block cipher), ChaCha20 (stream cipher)
  - Hash functions: SHA-2 & SHA-3 family, with digest size of 256 bits or higher

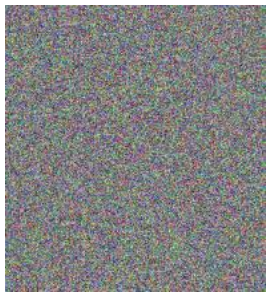
Block Cipher Modes of Operation

- The **mode of operation** specifies how to repeatedly apply a block cipher algorithm to encrypt a message which is larger than the size of one block
- **Electronic Code Book (ECB)** divides the message into blocks and encrypts each block separately. Not recommended (see **(b)** on the right)
- **Cypher-Block Chaining (CBC)** XORs each block of plaintext with the previous block of ciphertext before encrypting it
- **Counter (CTR)** turns a block cipher into a stream cipher

(a) Original image



(b) Encrypted with ECB mode



(c) Other modes (e.g., CBC) result in pseudorandomness

Source of images: Wikipedia. Picture of Tux (Linux mascot) created by Larry Ewing.

Symmetric Encryption: CBC Mode of Operation

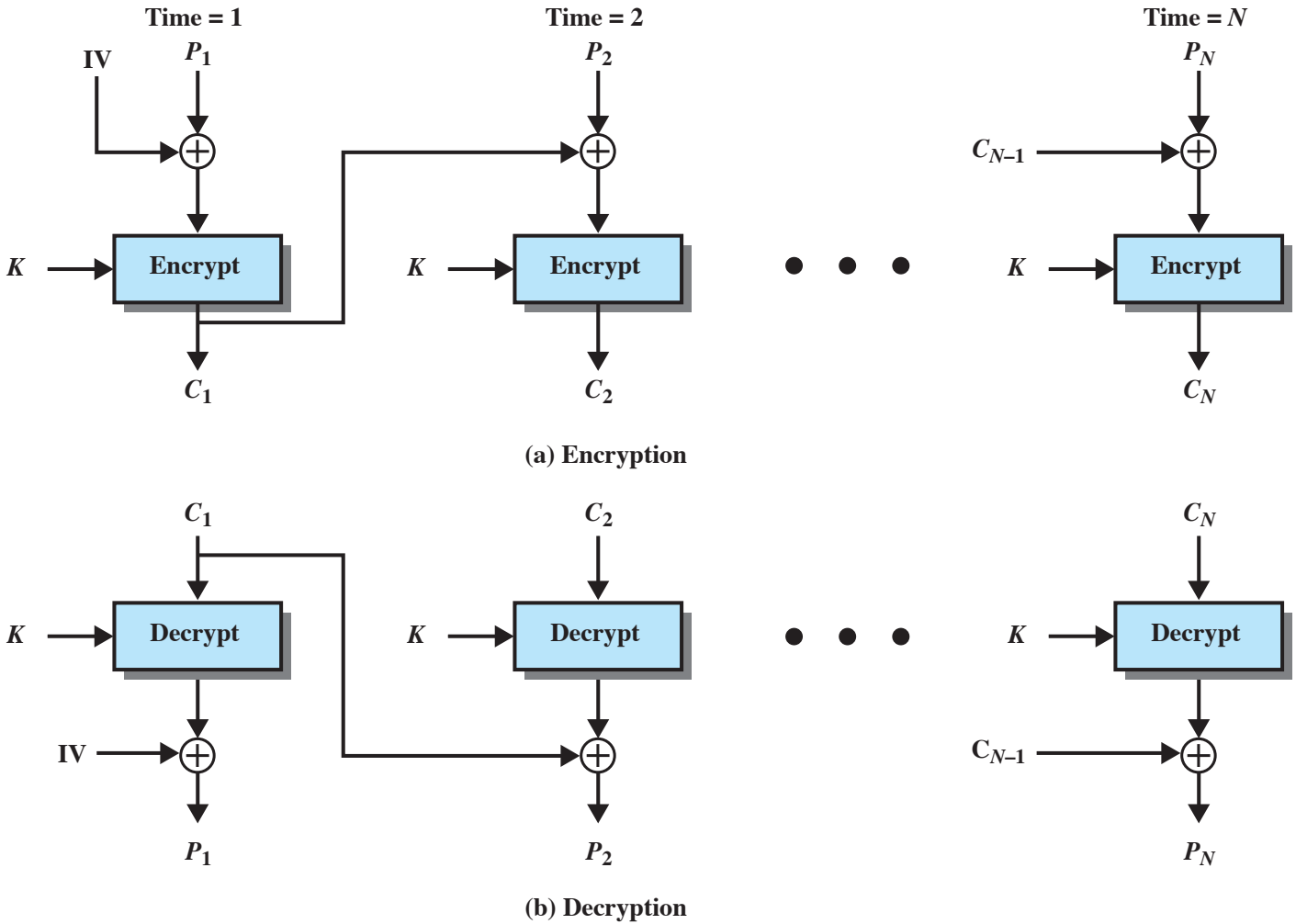


Figure 20.6 Cipher Block Chaining (CBC) Mode

Symmetric Encryption: Exhaustive Key Search

Time Required for Exhaustive Key Search (for Symmetric-key Encryption)

Key size (bits)	Approximate Number of Possible Keys	Time Required by Typical PC	Time Required by Supercomputer
56	10^{16}	1.125 years	1 hour
128	10^{38}	10^{21} years	10^{17} years
256	10^{77}	10^{60} years	10^{56} years

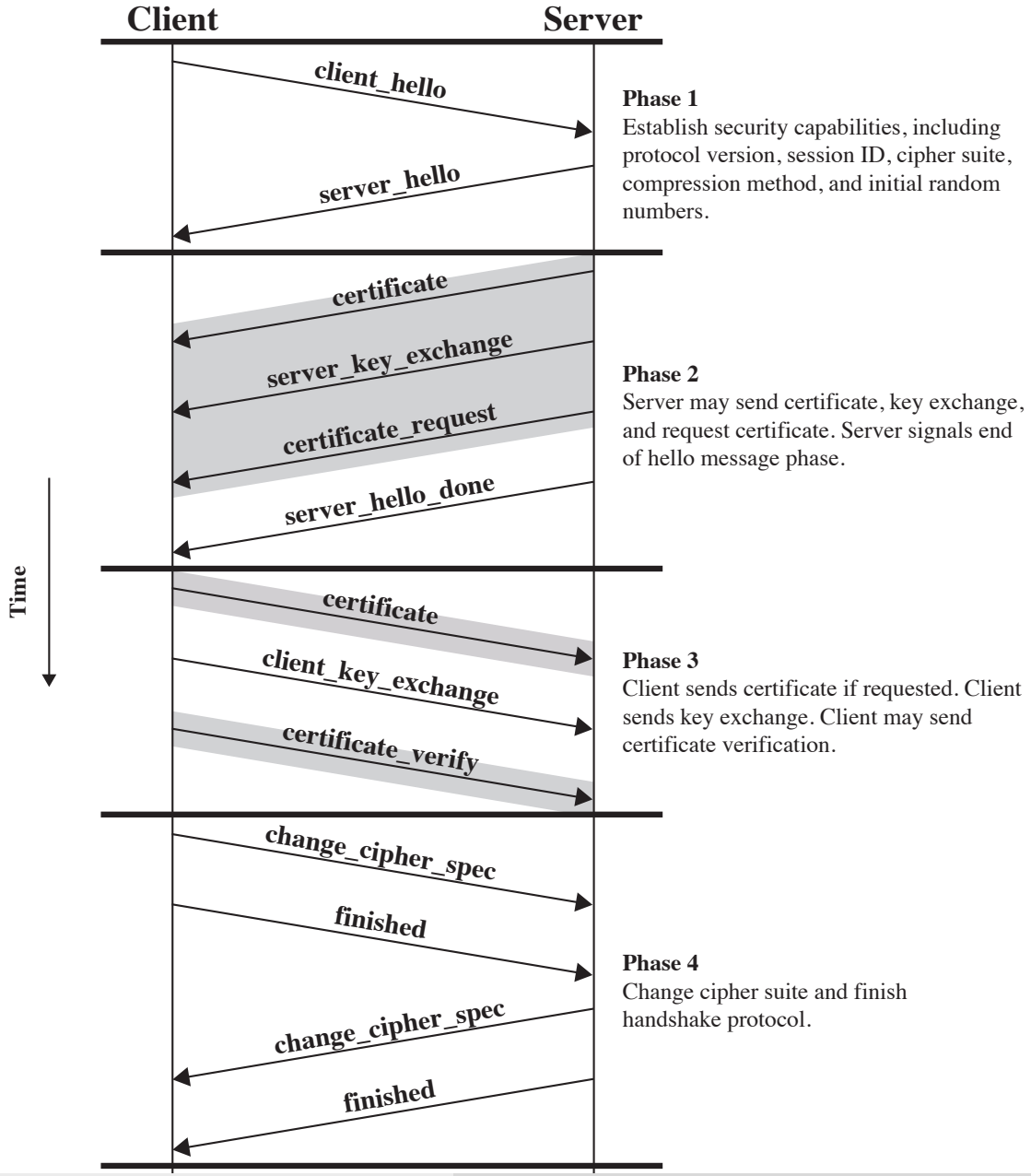
Age of the universe: 13.8 billion (1.38×10^{10}) years

TLS

- Transport Layer Security (TLS) provides reliable end-to-end secure service over TCP
- Successor to Secure Sockets Layer (SSL)
- HTTPS is the most popular application-layer protocol built on TLS
- Most recent version: TLS 1.3 (approved by IETF in March 2018)
- Uses public-key cryptography for **server authentication** (**client authentication** is supported, not commonly used) and for **key exchange**
- Uses symmetric-key cryptography for encrypting application-layer data

TLS Handshake Protocol

- Initiates the TLS connection
- Allows the server and client to:
  - Authenticate each other
  - Negotiate encryption and MAC algorithms
  - Negotiate cryptographic keys to be used
- Precedes any exchange of application-level data
- TLS 1.3 handshake reduced to 1 round-trip

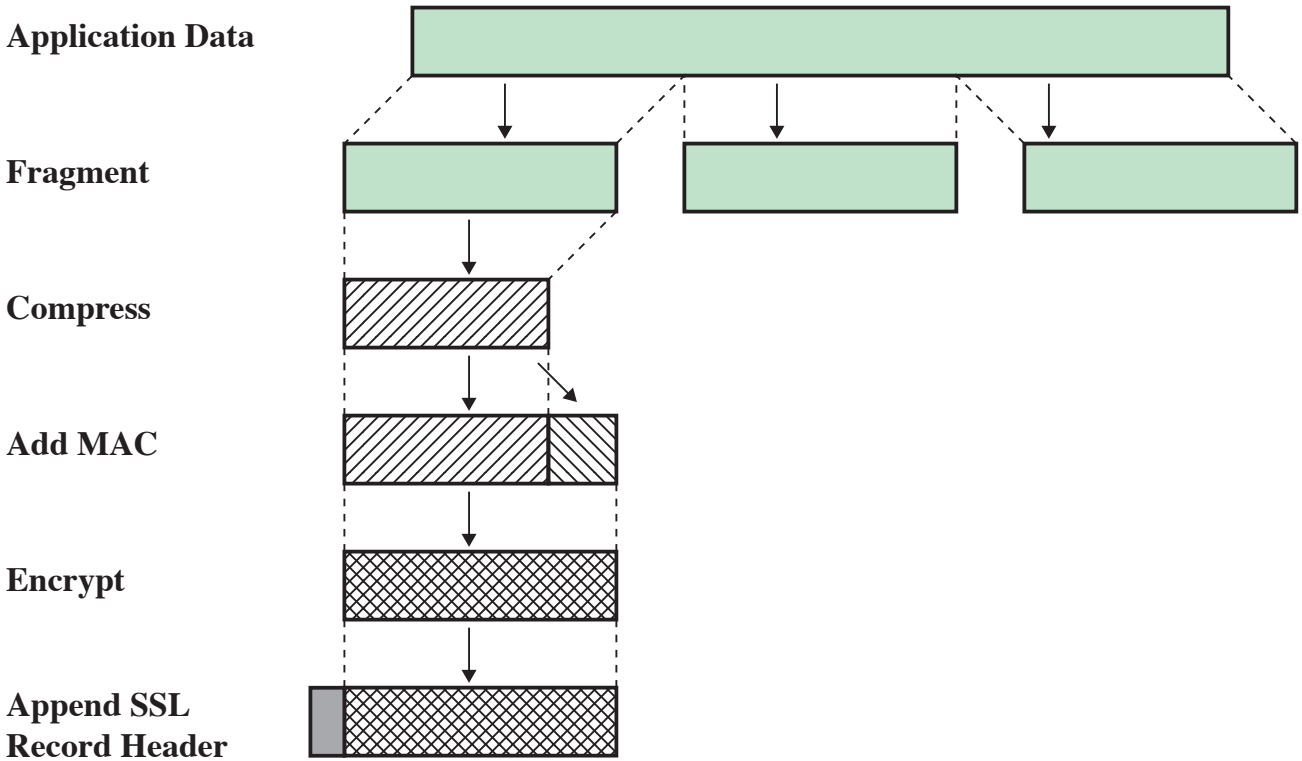


Diffie-Hellman Key Exchange

- Provides **perfect forward secrecy:** If an eavesdropper records all network traffic, they will be unable to decrypt the data even in the event of a future compromise of the server's private key
- Simple example to illustrate the concept (adapted from Wikipedia):
  - Alice and Bob agree to use a prime number $p = 23$ and base $g = 5$
  - Alice chooses a secret integer $a = 6$, and sends Bob:
    $$A = 5^6 \mod 23 = 8$$
  - Bob chooses a secret integer $b = 15$, and sends Alice:
    $$B = 5^{15} \mod 23 = 19$$
  - Alice computes the shared secret:
    $$s = 19^6 \mod 23 = 2$$
  - Bob computes the shared secret:
    $$s = 8^{15} \mod 23 = 2$$
- In practice, variations of the Diffie-Hellman key exchange are used which digitally sign the exchanged messages in order to protect **integrity** and **authenticity**
  - No need to protect **confidentiality** of the messages, since the shared key is never transmitted over the network

TLS Record Protocol

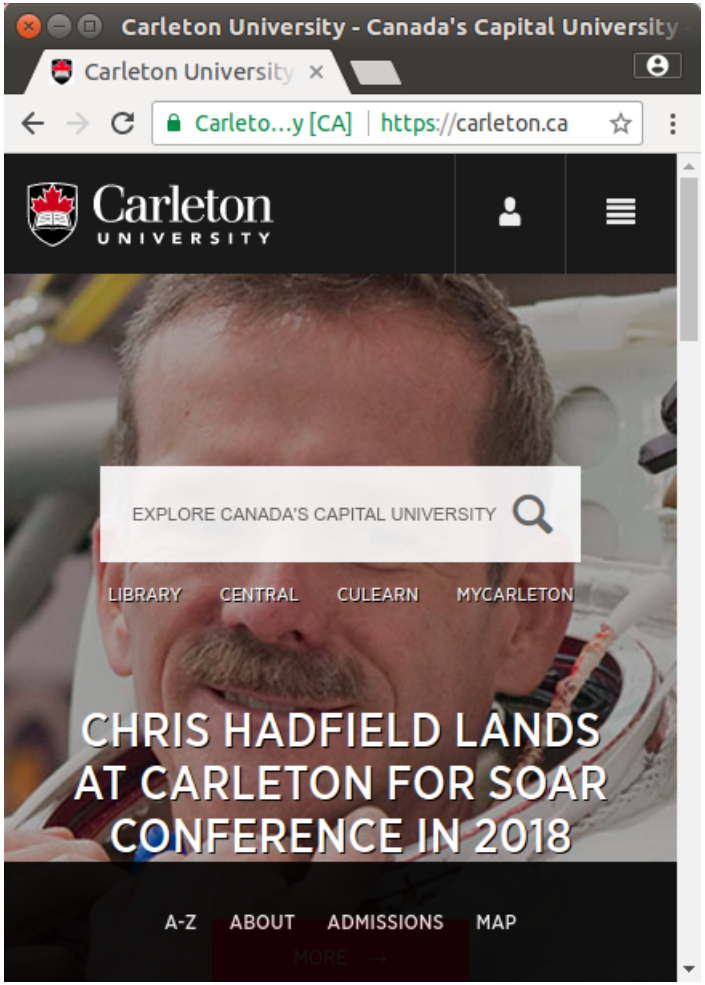
- Encapsulates application-layer protocol packets
- **Confidentiality** is provided by symmetric encryption of all application-layer data using a shared secret key
- **Message integrity** is provided by a MAC, which uses a separate shared secret key



Key Distribution: Web Browsing

- Key exchange between two communicating parties can be done by:
  1. Meeting in person to exchange or validate keys
  2. Relying on a trusted third-party to certify and validate keys

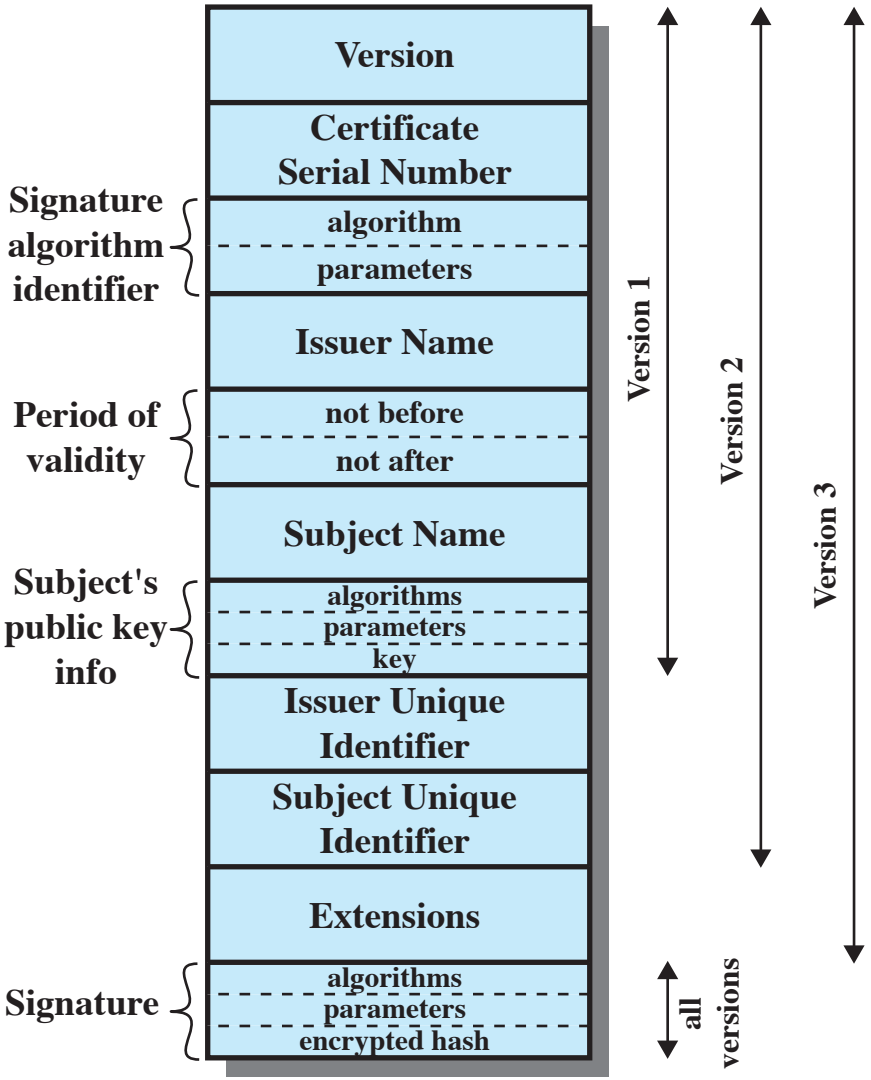
- For secure web browsing (HTTPS), public keys are certified (i.e., digitally signed) by **certificate authorities**



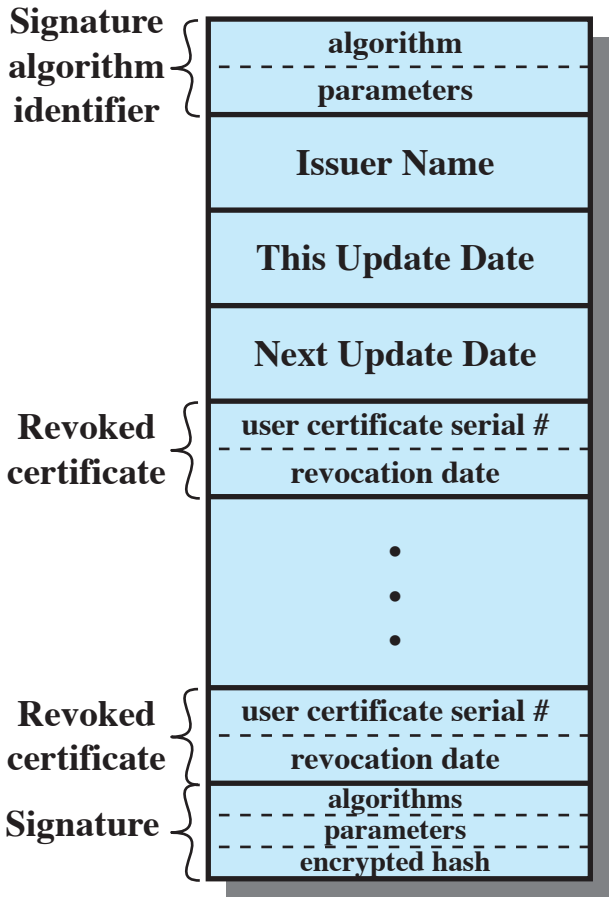
X.509 Certificates

- SSL/TLS (and many other protocols, e.g., IPsec) use X.509 public-key certificates for authentication
- Typically signed by a **Certificate Authority** (CA), which is a trusted third-party whose public key is pre-installed in the operating system or web browser
- A CA may also sign a certificate which designates the entity to act as an **Intermediate CA**
- The subject wishing a certificate provides their public key and any other required fields in the certificate, and present it to the CA to be signed
- The subject may wish to **revoke** their certificate:
  - In the event of a key compromise
  - To upgrade to a larger key size
- It is up to the client to check whether or not a certificate has been revoked: Often neglected in practice

X.509 Certificates: Structure



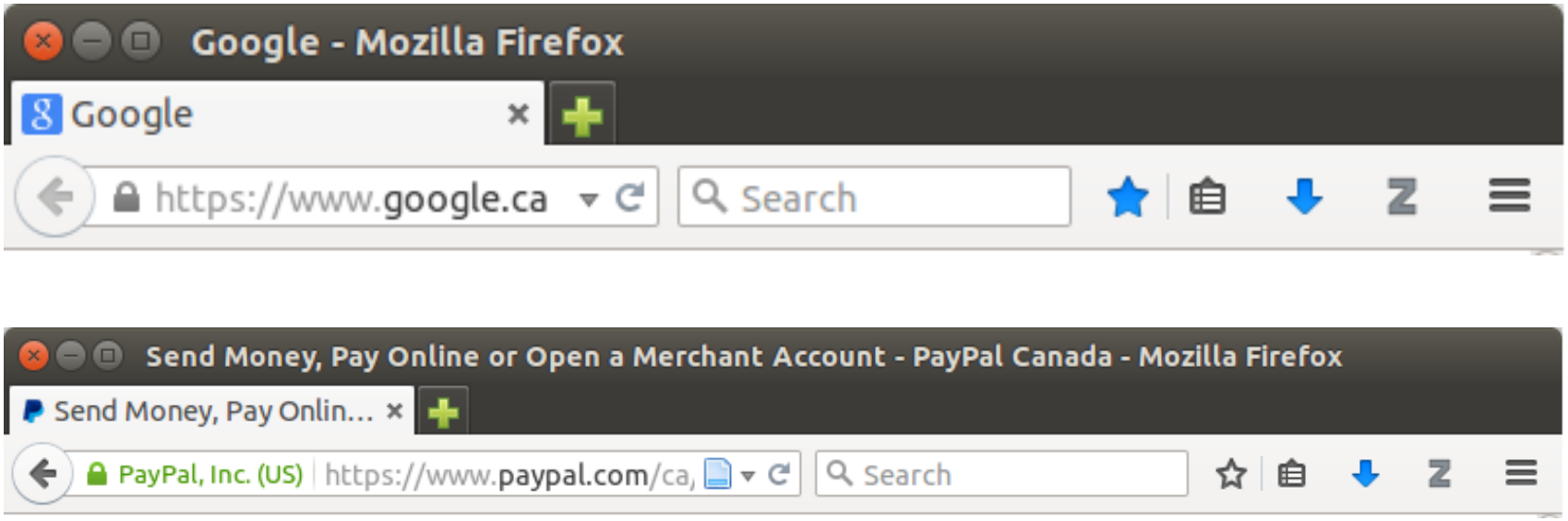
(a) X.509 Certificate



(b) Certificate Revocation List

Browser Cues: Domain vs. Extended Validation

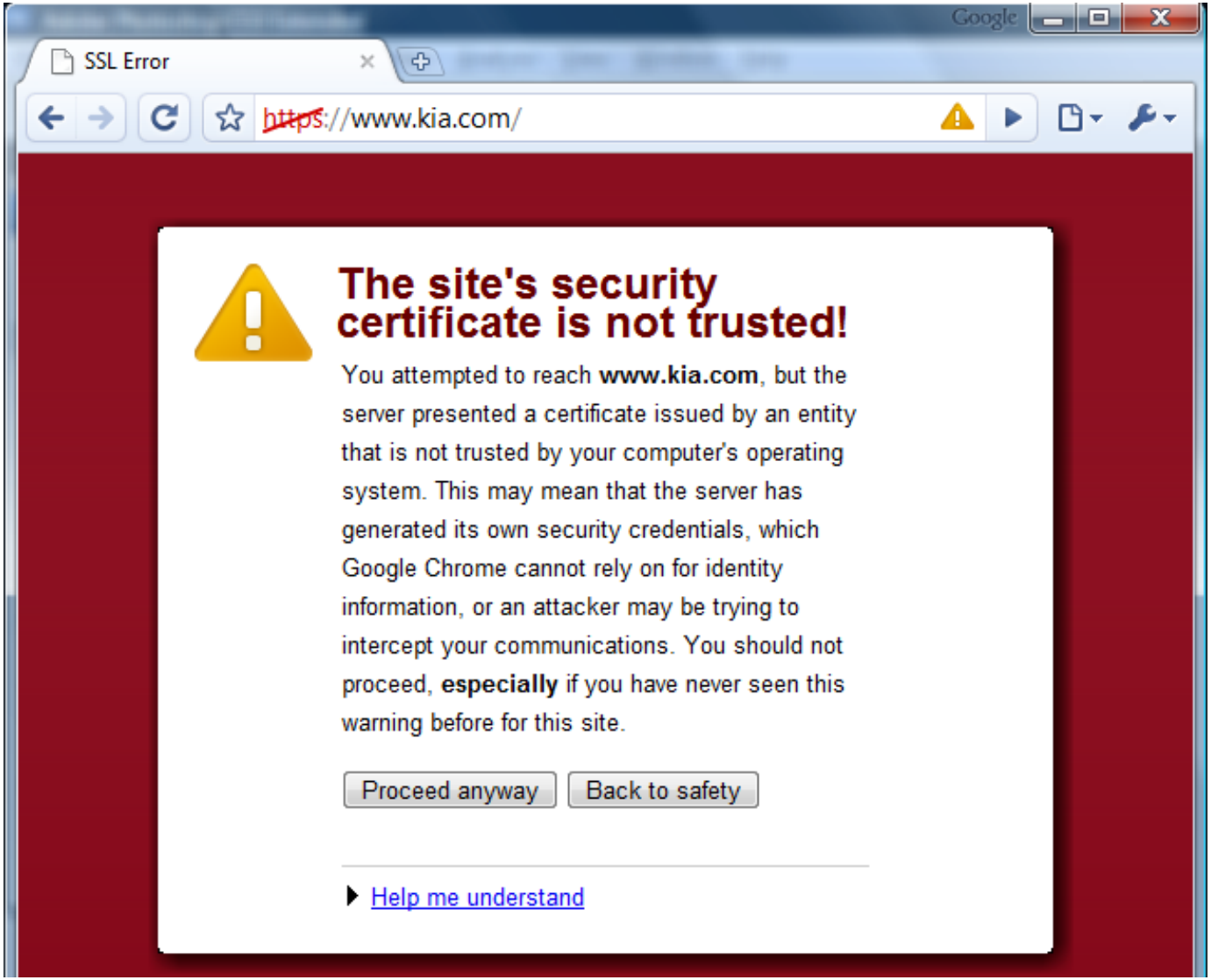
- A domain-validated certificate proves that the web server presenting the certificate is the legitimate owner of the domain specified in the certificate
  - CA typically verifies by sending an e-mail to an admin e-mail address associated with the domain name
- Extended validation certificates can only be issued by CAs who have demonstrated their adherence to a strict methodology for how they confirm the subject's identity



Man-in-the-Middle Attacks

- Man-in-the-middle (MITM) attacks involve an attacker which actively relays messages between two hosts, while making them believe that they are communicating directly with each other
- Requires the attacker to be able to intercept messages passing between two hosts, e.g., on an unencrypted WiFi network or a compromised router/gateway
- Basic approach for attacking a web browsing session: SSL stripping
  - Very easy to do
  - Users often do not notice the absence of security indicators
- More complex approach: Use a forged or compromised certificate
  - Requires attacker to know the server's private key, or to produce a fraudulent certificate signed by a trusted CA
  - User will still see HTTPS indicators

Browser Cues: Certificate Errors



The web browser can detect certificate errors (e.g., expired, invalid, not signed by a trusted CA, revoked) but typically gives the option to the user of proceeding anyway.

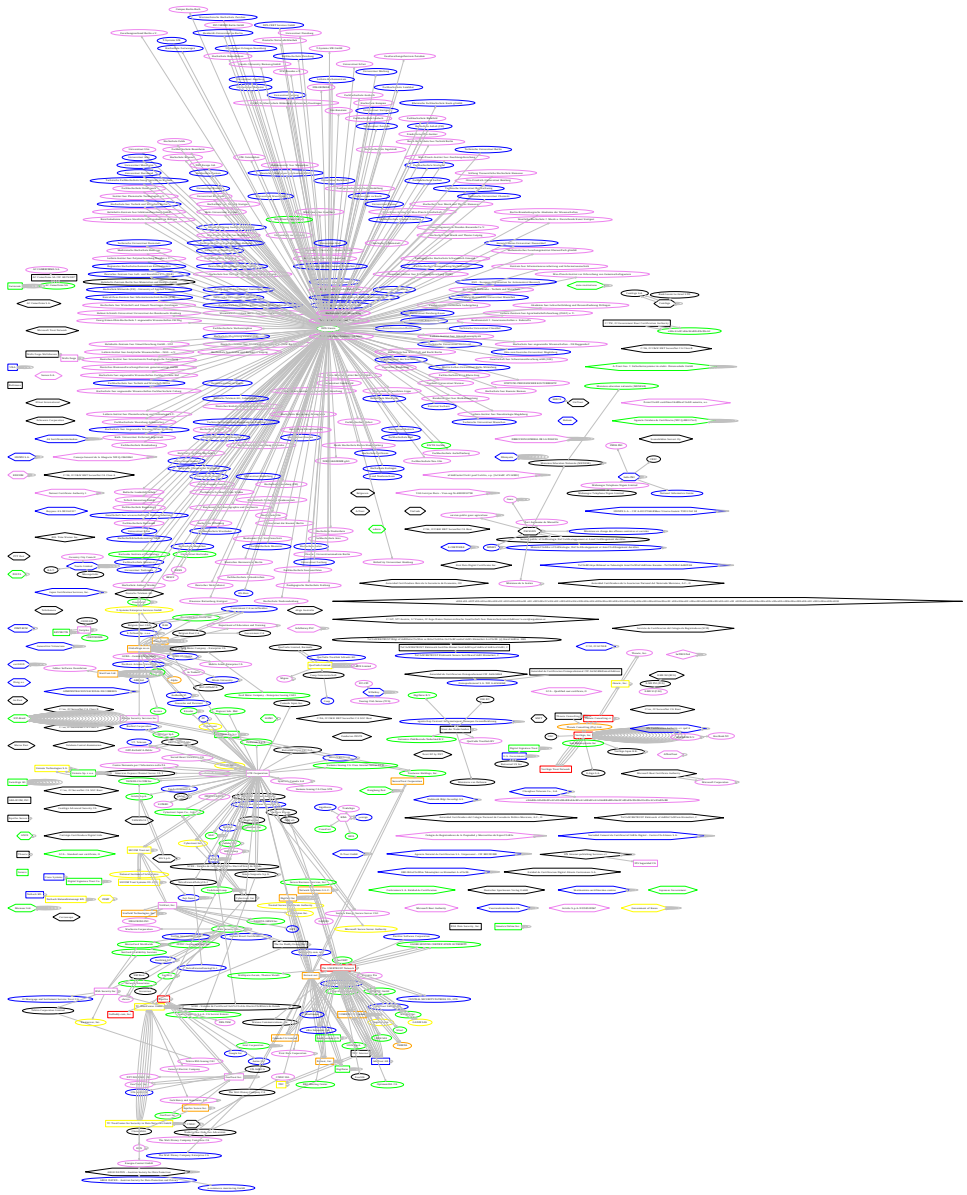
Crypto Attack: Hash Collisions

In Feb. 2017, Google used the "SHAttered" attack to construct 2 distinct PDF files that have the same SHA-1 hash. The attack required the equivalent of 1 year of computation time with 110 GPUs. The attack is about 100,000 times faster than a brute-force attack.

MD5 has long been known to be insecure, but check out this blog post from Oct. 2014 by Nathaniel McHugh, showing how he modified the below two images to produce the same MD5 hash: It cost 65 cents for 10 hours of computation time on an Amazon Web Services GPU instance.

Source: `http://natmchugh.blogspot.ca/2014/10/how-i-created-two-images-with-same-md5.html`

Graph of 650 CAs Trusted by Mozilla and Microsoft



Source: EFF SSL Observatory

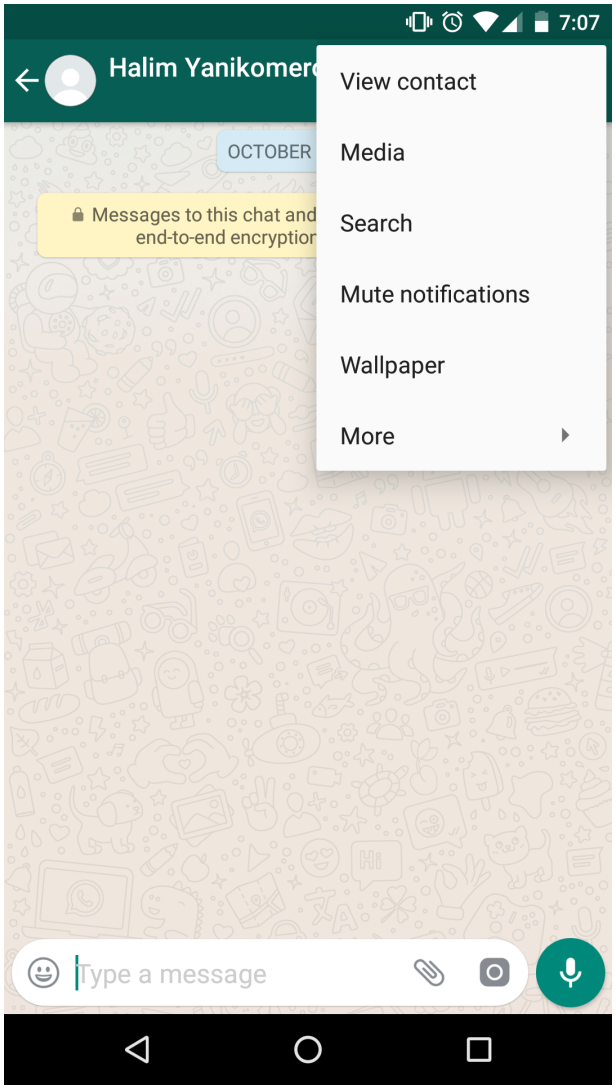
Public-Key Infrastructure (PKI) Challenges

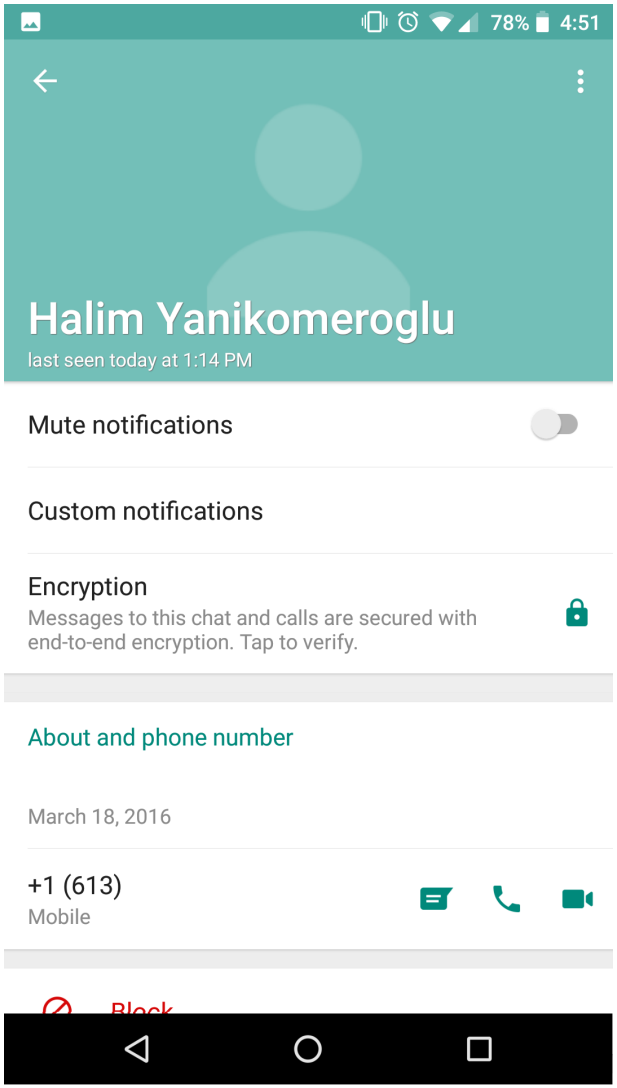
- PKI: Set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on public-key cryptography

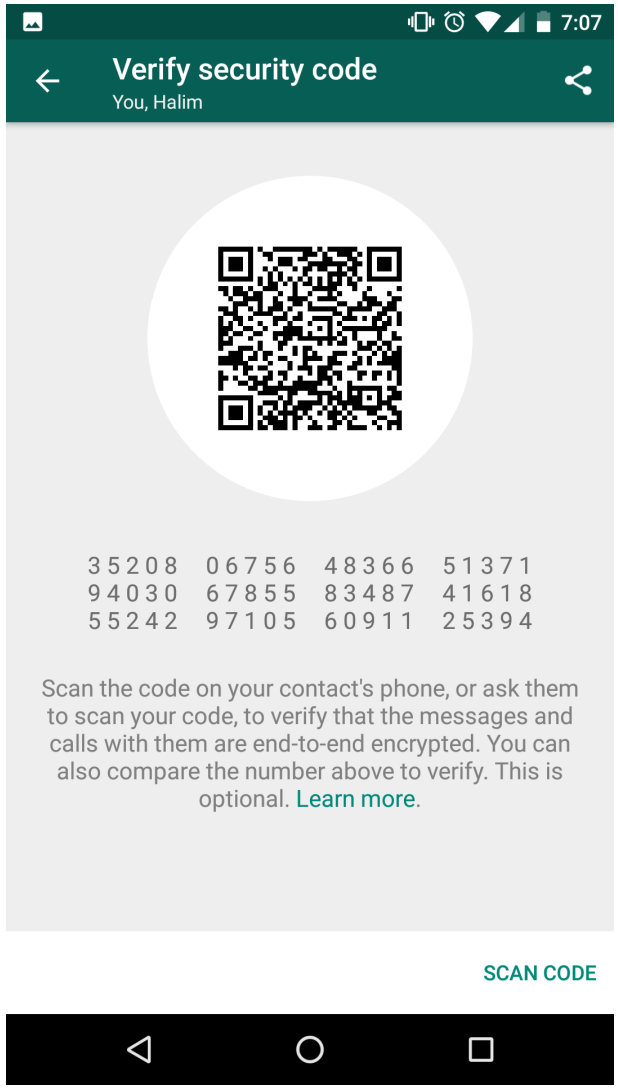
- PKI challenges:
  - Reliance on users to make an informed decision when there is a problem verifying a certificate
  - Assumption that all CAs in the "trust store" are equally trusted, equally well managed, and apply equivalent policies
  - Different "trust stores" in different browsers and OSs

- Some recent proposals & standards:
  - HTTP Strict Transport Security (HSTS)
  - Certificate pinning
  - Certificate transparency
  - Perspectives/Convergence
  - DANE

Key Distribution: WhatsApp







Figures credit & Other talks

- Figures/Tables from slides 8, 19, 11, 13, and 16 sourced from Computer Security Principles and Practice 3e by Stallings & Brown (2014)
- Interesting talks:
  - Lessons from Surviving a 300Gbps DDoS Attack (Matthew Prince, Black Hat 2013 talk)
    - `https://www.youtube.com/watch?v=w04ZAXftQ_Y`
  - The History and Evolution of Computer Viruses (Mikko Hypponen, DEFCON 2011 talk)
    - `https://www.youtube.com/watch?v=L8lA1pNvcz4`
  - SSL and the Future of Authenticity (Moxie Marlinspike, Black Hat 2011 talk)
    - `https://www.youtube.com/watch?v=Z7Wl2FW2TcA`