PETs - test 2 - summary

Contents

Secure Messaging **2**
Concepts 2
General methods 3
Message-based protocols 3
Pretty Good Privacy (PGP) 3
Session-based protocols 6
(OTR) Off-the-record messaging 6
Secure Mobile Messaging 6
properties of secure messaging 7
ephemeral messaging 7
Threema & iMessage (PGP) 8
messengers with forward secrecy 8
Re-decentralization 9
Anonymity and secure messaging 9
Tor messenger 9
Ricochet 10
Current events 10

Web Privacy **10**
Network Leaks 10
Domain Name Service (DNS) Leaks 10
HTTP(S) Leaks 11
Web Tracking 11
Online Advertisement 12
Social Networks and CDNs 12
Types of identifiable tracking-information 12
Tracking Technologies 13
Tracking Protection 13
Opt-out initiatives - industry self regulation 14
Browsers 14
adblock usage worldwide 15
adblock detection 15
Beyond the Desktop 15
mobile privacy 15
cross-device tracking 16
mobile privacy tools 16

TLS - Transport Layer Security **17**
Overview 17
Goals of TLS 17
Goals of TLS 1.2 17
TLS / PETS 17
TLS protocols 17
PKI - public key infrastructure 19

X.509 19
chain of trust 19
root CAs, trust stores 19
Implementation 20
OpenSSL problems 20
Cryptographic primitives 20
Ciphersuites 20
Application of TLS 21
HTTPS 21
TLS for Email 21
Incidents, Attacks & Flaws 22
Incidents 22
Implementation bug: Heartbleed 22
Crypto 22
Protocol Flaws 23
Other TLS attacks 23
Improvements 23
HSTS 23
Pinning 23
HPKP 23
CAA 23
Certificate Transparency 24
Let's Encrypt 24
HTTPS Everywhere 24
DANE 24
TLS 1.3 24
Major differences 24

Secure Messaging

Concepts

- Synchronicity
- Forward / backward secrecy
- Deniability

Synchronicity

- Synchronous:
  - Participants have to be online at same time
  - not feasible for many use cases
- Asynchronous:
  - third party caches messages
  - store and forward

Forward Secrecy

- feature of key agreement
- session key not compromised if private key compromised
- protects past sessions against future compromises

Plausible Deniability

- ability to deny knowledge/sending of message

General methods

- message-based protocols (PGP)
  - asynchronous long-lived message exchange
  - no forward secrecy
  - no plausible deniability
- session-based protocols (OTR)
  - synchronous ephemeral message exchange
- hybrid protocols (Signal)
  - asynchronous ephemeral sessions

Message-based protocols

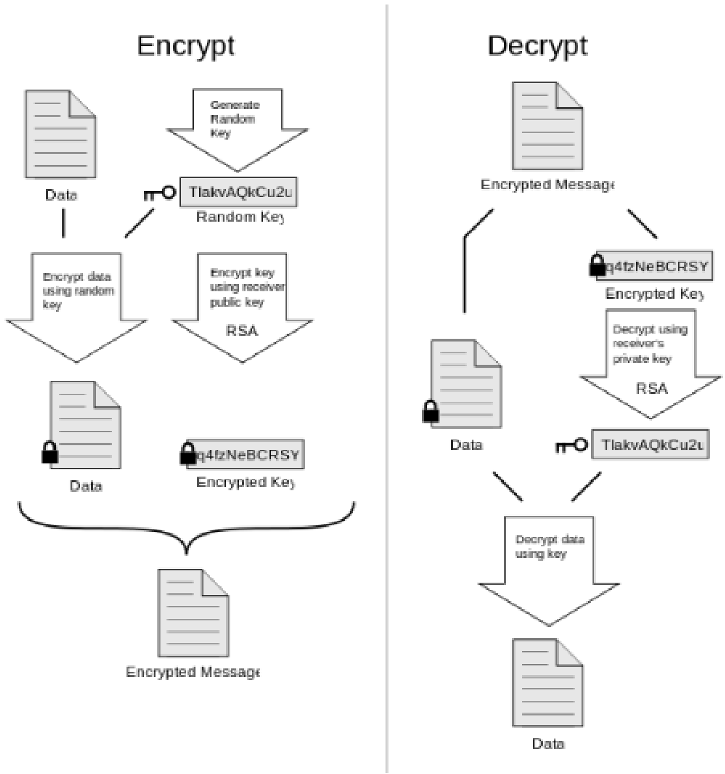
Pretty Good Privacy (PGP)

History

- first version: 1991 - Phil Zimmerman
- encryption & signing of files/emails
- first widespread use of public-key crypto

Functionality

- encryption
  - random key for symmetric encryption, key then encryptet with public key of recipient
- decryption
  - recipient uses own private key to decrypt message key
- signing
  - cryptographic hash of message signed with private key of sender
- authentication
  - recipient validates encryptet hash with public key of sender



public PGP key

- public key on personal website
- public key servers
- https://keybase.io - linked to social media account
- fingerprint of public key
  - hash of public key in HEX
  - short ID: last 8 chars of fingerprint

Verification of public keys Web of Trust

- signing of other PGP user’s public keys
- keys with more signatures ratet more trustworthy
- signatures from people with multiple signature count more
- key-signing-parties

S/MIME

- hierarchical PKI
  - compare to TLS
  - in contrast to web-of-trust
- get trusted certificate, e.g. from TU

PGP Software

- PGP Corporation
- GNU privacy guard (GnuPG/GPG)
  - open-source implementation of OpenPGP standards
  - GPG as such is a commandline tool

Advantages of GPG/PGP

- strong end-to-end encryption
- hybrid encryption
  - encryption with fast symmetric ciphers, random password
  - enc. password protected with asymmetric ciphers
- good software support

Disadvantages of PGP

- no forward-secrecy
  - attacker collects encrypted emails
  - once new attacks available / private key stolen
  - previous messages can be decrypted
- no plausible deniability
  - messages signed with private key of sender

Usability vs. PGP

- Why Johny can't encrypt
  - survey based on PGP 5.0
  - a lot of misunderstanding regarding use of PGP
  - e.g. people distribute private keys to communicate
- replies to encrypted e-mails in plaintext
- usability breaks PGP security model

General Problems

- people lose private keys / do not use it
- privacy issues
  - web of trust: personal social network becomes public
  - metadata not protected

Session-based protocols

(OTR) Off-the-record messaging

- primary application: internet chats
- supports:
  - encryption
  - authentication
  - perfect forward secrecy
  - plausible deniability
- combination of:
  - AES
  - Diffie-Hellman
  - SHA-2 hash

perfect forward secrecy

- New AES key for every exchanged message
  - exchange via ephemeral diffie hellman keys
  - ephemeral keys signed with long term keypair

Deniability

- authenticity via MAC (Message Authentication Codes)
- previous MAC key published with next message (everybody can fake old message)

Using OTR

- can be used with most common chat protocols
- native support or plug-ins
- limitations:
  - group-chats
  - support for multiple devices
  - asynchronous communication

Secure Mobile Messaging

- “Snowden effect”
  - general awareness for privacy on the rise
  - number of new tools for general public / companies
  - “military grade encryption”

properties of secure messaging

- first suggested properties
- out of date, more to consider
  - client-server encryption
  - end-to-end encryption
  - trust/FP verification
  - forward secrecy
  - open source
  - design documentation
  - recent code audit

client-server encryption

- encrypt communication in transit
- protection against simple eavesdropping attacks
- plaintext at service provider
- provider can read & share messages
- mostly TLS used
  - introduces all problems of TLS
  - verification of certificates
  - pinning of certificate

end-to-end encryption

- provider unable to read messages
- only clients can decrypt
- e.g. PGP encryption
- other possible protocols (e.g. Signal)

Contact verification

- how to verify contacts?
- authentication mechanism
- usage without phone number / email

ephemeral messaging

ephemeral: *lasting for a very short time*

- messages deleted after some time
- time-out setting for conversations
- example: snapchat
- client deletes photos (trust in client device)
- Secret / Whisper / Snapchat / etc.
  - messages temporarily saved on device
  - little information on storage duration on server
  - provider can read all messages

    - deceptive marketing

Threema & iMessage (PGP)

- Threema
    - entropy generated with user input
    - simple “traffic light” system, verification via QR-code
    - PGP (no perfect forward secrecy)
- iMessage
    - standard PGP over XMPP
    - easy to use
    - keys might be store in cloud
    - PKI infrastructure under control of Apple

messengers with forward secrecy

- Telegram
    - popular WhatsApp alternative
    - MTProto protocol (controversial)
    - 2 different encryption modes
    - default: client-server encryption
    - end-to-end encryption
        - has to be manually activated, contact needs to be online
        - authentication only face-to-face
- Signal
    - first version based on OTR protocol
    - initially for SMS messages
    - version 2.0
        - internet-based exchange
        - optional sms fall-back
        - protocol now used in WhatsApp & Facebook Messenger

double ratchet algorithm

- introduced als *axolotsl protocol*
- combines
    - DH ratchet from OTR
    - symmetric-key ratchet from SCIMP
- new key for each message
- core concept: key derivation function chain

Signal protocol

- double-ratchet algorithm
- 3DH key exchange
- prekeys
- EC25519, AES256

Signal - discovering other users

- discover friends in privacy-preserving way
  - hard problem
  - contact data hashed, sent to server for comparison
  - hash of phone number useless
- encryption bloom filter
  - no contact data sent to server
  - encrypted bloom filter with all contacts queried locally
- new contact discovery (2017)
  - SGX service - remote attestation

Re-decentralization

- PGP/GPG: decentralized
- OTR for e.g. XMPP: decentralized
- mobile messaging: centralized
- matrix: decentralized

Matrix / riot.im

- open-source specification
- HTTP APIs
- federated messaging
- riot.im: client, reference implementation
- demand for interoperable applications?

Anonymity and secure messaging

metadata is the name of the game, and e2e encryption the honeypot

- all introduced applications offer confidentiality but metadata is leaked
- provider metadata and/or traffic analysis

Tor messenger

- cross-platform messenger
  - support number of protocols: Jabber/Google Talk / FB messenger, etc.
  - transport automatically via Tor
  - OTR enabled by default
- still possible to force providers for communication logs

Ricochet

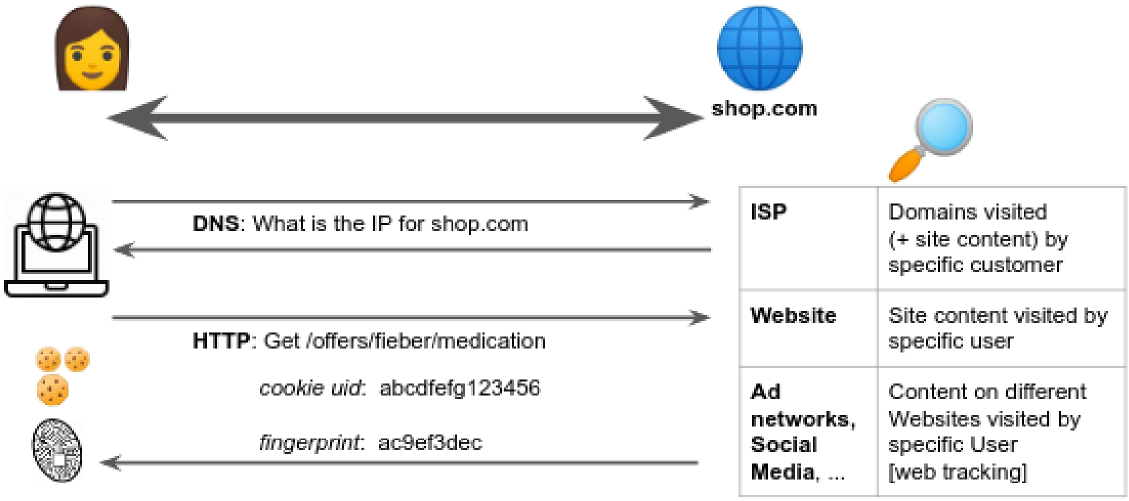
- *anonymous instant messaging for real privacy*
  - builds upon Tor hidden services
- no central server
- custom binary messaging protocol
- user name: ricochet:. . . . .
- uses encryption already available through Tor

Current events

- politicians urge for crypto backdoors
- intelligence agencies are “going dark”
  - metadata available in majority of cases
  - backdoors make products insecure for everyone
  - targeted attacks always possible

Web Privacy

Network Leaks


shop.com
DNS: What is the IP for shop.com
HTTP: Get /offers/fieber/medication
cookie uid: abcdfefg123456
fingerprint: ac9ef3dec

ISP	Domains visited (+ site content) by specific customer
Website	Site content visited by specific user
Ad networks, Social Media, ...	Content on different Websites visited by specific User [web tracking]

Domain Name Service (DNS) Leaks

- DNS is plaintext protocol (UDP port 53)
- Requests visible within WiFi, to ISP, in transit
- monitoring independent of DNS provider
- security: DNS response spoofing (censorship, advertising via hijacking)

Encrypted DNS

- DoT: DNS wrapped with TLS (new port tcp 853)
  - potential issue: blocking / detection
  - supported on Android, systemd on Linux
- DoH: HTTPS for transporting DNS queries
  - HTTPS commonly used for web services / browser APIs
  - supported by Chrome, Firefox, Opera

Discussion around encrypted DNS

- ISPA criticized Mozilla & Google for adapting DoH
  - undermining blocking lists
  - blocking + monitoring still possible
- Mozilla defaults to CloudFlare's DNS when enabling DoH
  - CF can link requests to source IP / user agents

HTTP(S) Leaks

- unencrypted HTTP
  - websites requested http://shop.com/xyz/abc/def
  - entire page content including authentication token
  - straightforward to monitor with transparent http proxies
- HTTPS
  - hostname leaks in initial TLS handshake
  - deep packet inspection used to monitor / censor HTTPS

Web Tracking

- web tracking = creation of unique user profiles
- **first parties**
  - websites
  - mobile application
- **third parties**
  - advertisement
  - analytic providers
  - online social networks
- *trackers* link people to sensitive information

Online Advertisement

- direct sales
  - links to products on websites / social media (usually no tracking by third paries)
- Ad networks: place ads on multiple websites, targeting ads based on:
  - demographics
  - location based
  - website content
  - user profiles
- Ad exchanges
  - auction of available advertisement spaces
  - sell customer information

Social Networks and CDNs

- social plugins aka. *share buttons*
  - single-sign-on
  - shreThis, Addthis → collect user information
- content provider
- javascript libraries
- webhoster

Types of identifiable tracking-information

- third-party is also first party
  - users linked via Facebook-like-button with real name
- first party sells user data
  - personal information directly sold to e.g. ad networks
- unintentional sharing of personal information
- misuse of security bugs
  - XSS, clickjacking, history stealing
- re-targeting
  - e.g. match users by profile pictures

Tracking Technologies

- tracking via third-party libraries
  - visited URL leaked via referer or submitted directly
- user profiles: HTTP tracking cookie
  - unique cookie, set on initial loading of website
- supercookies
  - multitude of storage location for user identifier except HTTP cookie
  - use alternative storage locations
  - cookie resyncing (restored from one of many supercookie storage locations)
- fingerprinting
  - tracking via unique OS/browser properties
  - persistent tracking of users without cookies
  - based on unique system properties

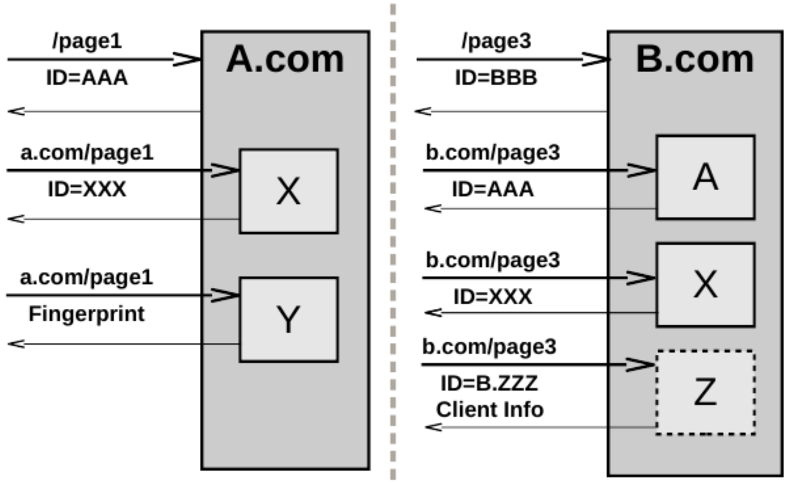


Figure: **A** ... First- and Third-party (e.g. Facebook), **X** ... advertisement network (e.g. doubleclick), **Y** ... uses fingerprints instead of cookies, **Z** ... analytics service (e.g. Google Analytics)

Tracking Protection

- website providers
  - same-origin policy (dedicated websites for tracking)
  - AnonymizeIp or e.g. Matomo
  - alternatives to standard social plugins
- opt-out
  - = no target advertisement
  - privacy initiatives by industry
  - trust issue: how is data handled?
- brwoser settings / extensions
  - settings & features in current browsers
  - special browser extensions

Opt-out initiatives - industry self regulation

- special websites to set opt-out cookies
  - issues: validity / deletion of cookies, trust
- browser extensions for persistent opt-out cookies
- Do Not Track (DNT) HTTP header
  - up to websites to honer DNT header or not
  - was enabled by default → ignored

Browsers

- **Google Chrome**
  - advanced security measures (e.g. site isolation)
  - Google ad revenue = no anti-tracking
  - always sign-in first-party tracking across Google products
- **Safari**
  - intelligent tracking prevention 2.1
    - separate context for third-party cookies
    - purging of third-party cookies after 30 days
    - first-party cookies purged after 7 days
- **Firefox**
  - tracking prevention based on Disconnect ruleset
  - enhanced tracking prevention
  - multi-account containers
- **Brave**
  - tracking & fingerprinting protection
  - tor-browser tabs
  - “brave-rewards”: privacy-respecting ad ecosystem

browser settings

- deletion of cookies, cache
  - manual or once browser closed
  - supercookies survive
  - loss of settings & active sessions
- Do Not Track Header
- Third-party cookies
  - can be completely blocked
- private mode
  - no data locally stored

browser extensions

- Abblock Plus
  - most popular extension to block ads
  - ads blocked & set invisible
  - issue: acceptable ads (enabled by default)

- Ghostery
  - detection & blocking of web trackers
  - overlay for social plug-ins
  - issue: usability
  - issue: business model
- EFF Privacy Badger
  - based on heuristics
  - tests if DNT header honored
  - challenge: maintain whitelist
  - overlays for social plug-ins
- Disconnect.me
  - similar to Ghostery, but open-source ruleset
  - VPN service for mobile devices
  - basis for tracker blocking in Firefox
- uBlock (origin)
  - open-source “wide spectrum” blocker
  - focus on performance
  - challenge: overblocking, filterrule maintenance

adblock usage worldwide

- main motivation: security and annoyance
- asia: mobile browsers pre-configured with adblockers
- global: more educated users rely on adblockers

adblock detection

- baiting: inject (random) html-tag, check if blocked
- integrity checks: verify if certain scripts are loaded
- 75% of users leave websites with adblock detection

Beyond the Desktop

mobile privacy

- smartphone apps collect number of sensitive information
- third-party providers (ads, analytics, social SDKs)
  - access sensitive information
  - rely on unique device identifiers

cross-device tracking

- holy grail for marketers
  - profile shopping habits across multiple devices
- probabilistic methods
- big players
  - collect identifiers once authenticated with their SDKs
  - common third-parties in apps
- mew methods: e.g. SilverPush Audio beacons

ultrasonic beacons

- ultrasound out of human hearing range
- electronic devices play & receive ultrasound
- easy to encode data in ultrasound

mobile privacy tools

- Anti Web Tracking
  - iOS blocking extensions for Safari
  - Mobile Firefox + extensions
  - specialized privacy browsers: bromite, ghostery, etc.
- Extended protections that include tracking by apps
  - require rooting/jailbreaking
  - not feasible for average user

DNS

- DNS based blocking
  - reply to known tracking domain with `domain unknown`
  - course grained in comparison to browser extensions
  - `ads.facebook.com` can be blocked DNS-based
  - `facebook.com/ads` leads to overblocking
- using DNS blocking
  - specific android apps: DNS66
  - external services: special VPN, adblocking DNS resolvers
  - running own blocking DNS (e.g. Pi-Hole, upribox)

TLS - Transport Layer Security

Overview

Goals of TLS

- authentication
- confidentiality
- integrity
- TLS is application protocol independent

Goals of TLS 1.2

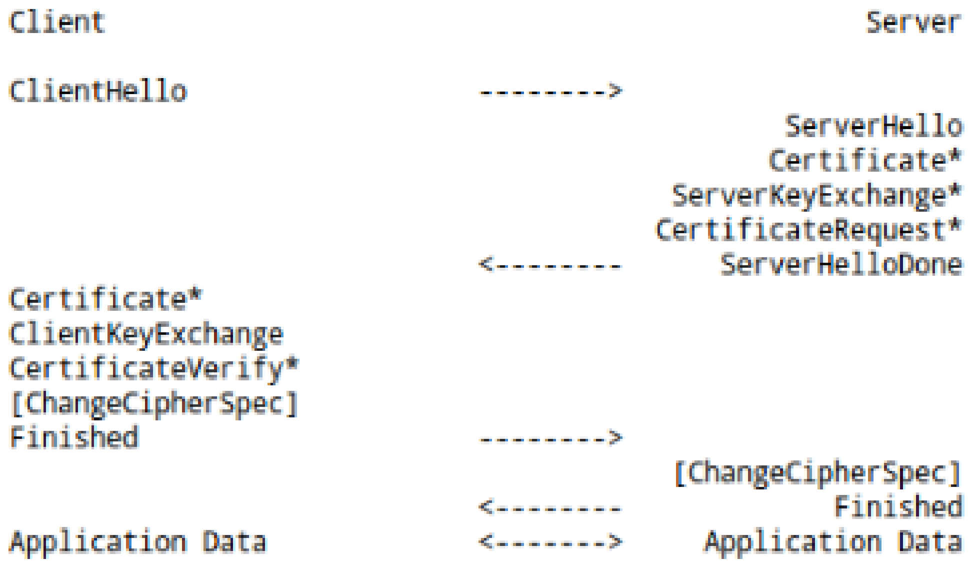
- cryptographic security
- interoperability
- extensibility
- relative efficiency

TLS / PETS

- **foundation of encrypted internet**
- improvements / incidents / vulnerabilities
- metadata not private
- no silver bullet for security

TLS protocols

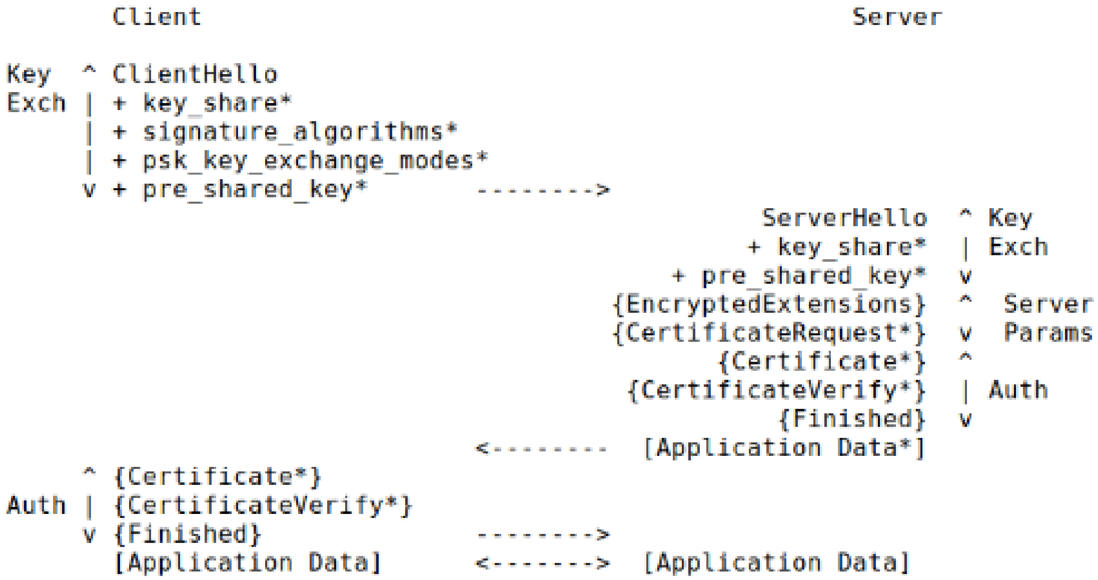
two primary concepts - handshake protocol - authenticates communicating parties - negotiates cryptographic modes - establishes shared keying material - record protocol - protect traffic between communicating peers



```
Client                                          Server

ClientHello                  -------->
                                           ServerHello
                                          Certificate*
                                    ServerKeyExchange*
                                   CertificateRequest*
                             <--------     ServerHelloDone
Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                     -------->
                                    [ChangeCipherSpec]
                             <--------        Finished
Application Data             <------->  Application Data
```

Figure 1: full handshake TLS 1.2



```
       Client                                           Server

Key  ^ ClientHello
Exch | + key_share*
     | + signature_algorithms*
     | + psk_key_exchange_modes*
     v + pre_shared_key*       -------->
                                                  ServerHello  ^ Key
                                                 + key_share*  | Exch
                                            + pre_shared_key*  v
                                        {EncryptedExtensions}  ^  Server
                                        {CertificateRequest*}  v  Params
                                               {Certificate*}  ^
                                         {CertificateVerify*}  | Auth
                                                   {Finished}  v
                               <--------  [Application Data*]
     ^ {Certificate*}
Auth | {CertificateVerify*}
     v {Finished}              -------->
       [Application Data]      <------->  [Application Data]
```

Figure 2: full handshake TLS 1.3

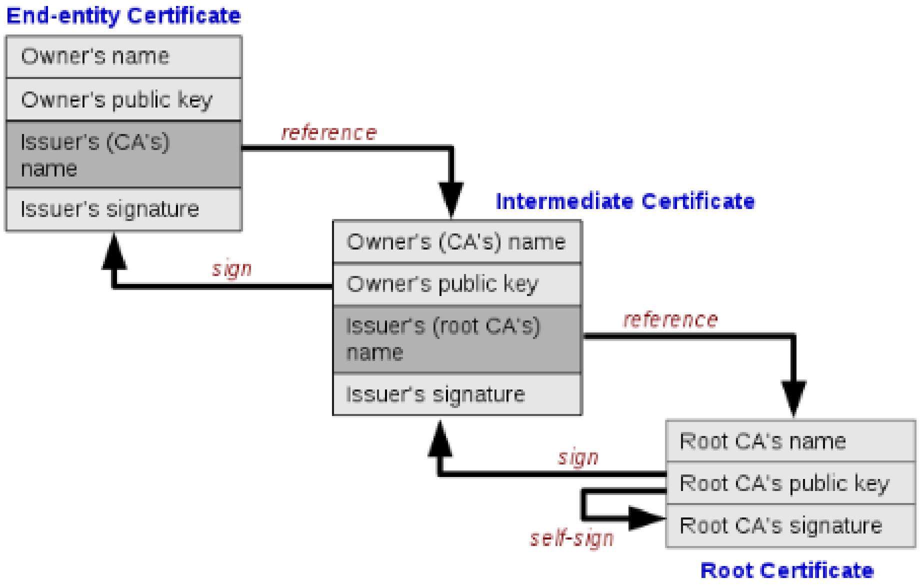
PKI - public key infrastructure

- certificates based on pubkey encryption
- CA issues certificates
- CA rights can be delegated: Sub-CA
- chain of trust to root CAs
- root CAs are trusted

X.509

- standard for pubkey certificates
- structured, e.g.
  - issuer name
  - subject name
  - validity
  - extensions
  - ...
- .pen /.crt / .cer / .der / not .csr / not .key / ...

chain of trust



root CAs, trust stores

- each browser & OS has set of trusted CAs
- CAs could sign everything
- not all signed HTTPS certificates
- controlled by different organizations, nations, ...
- 3 organizations control 75% of trusted certificates

Implementation

- OpenSSL: de-facto standard, swiss army knife
- LibreSSL: fory by OpenBSD team
- BoringSSL: Google
- GnuTLS
- NSS: Mozilla
- Microsoft Secure Channel
- s2n: Amazon
- miTLS: verified implementation

OpenSSL problems

- had own memory management, prevented many analysis tools
- bugs unfixed for long time
- code base unreadable
- extensive backward compatibility

Cryptographic primitives

Ciphersuites

- specifies cryptographic algorithms & modes
- consist of
  - key exchange
  - authentication
  - symmetric cryptography for transport
  - integrity (hash)
- server & browser support certain set
- negotiated while handshake
- key exchange:
  - DH
  - RSA for authentication
  - RSA issue: private key can decrypt prev. communication content
- foward secrecy:
  - DHE_RSA: ephemeral DH
  - ECDHE_RSA: elliptic curve DH
- encryption:
  - block ciphers: AES, 3DES, Camellia
  - or stream ciphers: RC4, ChaCha

TLS_DHE_RSA_WITH_AES_256_CBC_SHA

- DHE for key exchange
- RSA for authentication
- AES 256bit in CBC mode for encryption
- SHA for hashing

Application of TLS

HTTPS

- most widely used application layer protocol for TLS
- over 443

HTTPS problems

- HTTPS adoption
  - not used widely enough
  - use HTTPS not only for “high important” pages
  - certificates cost money
  - self-signed certificates bring problems
- secure deployment
  - complex task
  - e.g. correct ciphersuites
  - grading with SSLTest
  - hard to find good configuration
  - no secure defaults
  - bad documentation
  - lacking tool support
- usability
  - security for people
  - disruptive security concepts (browser warnings)
  - connection security indicators (icons)
  - admins should be seen as users too
- who leads the way?
  - browsers, CAs, service providers

TLS for Email

- dedicated TLS ports (465, 993, 995)
- STARTTLS to upgrade unencrypted connections
  - important for all email protocols: POP, IMAP, SMTP
  - ‘opportunistic encryption’ - if possible
  - does not defeat active attackers

Incidents, Attacks & Flaws

Incidents

PKI: DigiNotar

- CA from Netherlands, hacked 2011
- Fox-IT investigated attack
- DigiNotar bankrupt, removed from all browsers
- problems:
  - all signing servers in one AD, weak password
  - reachable over management LAN
  - no antivirus on servers
  - public webserver unpatched
- operation Black Tulip:
  - detected due to TLS pinning in Chrome
  - at least 531 fraudulent certificates issued
  - used to attack Gmail users MITM in Iran

MITM attacks

- most get detected with Chrome pinning Google certificates
- sometimes self-signed certificates

CAs distrusted

- 2016: Apple, Chrome, Mozilla distrust WoSign & StartCom
- multiple rule violations
- 2017: Google, Mozilla stop trusting Symantec certificates

Implementation bug: Heartbleed

- vulnerability in OpenSSL, 2014
- in Heartbleed protocol in TLS, missing bounds check
- up to 64kb readable form heap
- could contain user data, passwords, TLS private key

Crypto

Ps and Qs

- problem for creating pubkeys
- RSA chooses parameters at random
- for devices with low entropy collision possible
- problematic for embedded devices

Protocol Flaws

- DROWN
- POODLE

Other TLS attacks

- SMACK (State Machine Attacks)
- Logjam (Downgrage, Weak DH)
- FREAK (Downgrade, Factoring RSA export keys)
- CRIME, BREACH (HTTP compression)
- Lucky 13 (against CBC mode)
- . . .

Improvements

HSTS

- HTTP Strict Transport Security
- part of HTTP header response from server
- stores HTTPS preference
- error message instead of warning
- problem: TOFU (Trust On First Use)
- preload list

Pinning

- key distribution problem
- 'solved' with PKI, but PKI has problems
- pin certificate or pubkey (e.g. directly in browser/source code)
- not scalable

HPKP

- HTTP Public Key Pinning
- part of HTTP header response from server
- stores pinned key
- dead?
  - pin: leaf cert, intermediate cert or root cert
  - pubkey-pins-report-only
  - dead . . . planned removal in Chrome, 2018

CAA

- DNS record: Certification Authority Authorization
- *which CAs allowed to issue certificate for my domain?*
- mandatory for CAs since 2017
- CA check - not client system check

Certificate Transparency

- RFC6962
- logs: records of certificates
- logs: everyone could host, currently Google and CAs
- monitor: watch for suspicious certificates
- auditor: verify that logs behave correctly
- warning for certificates without CT log entry

Let's Encrypt

- free CA
- open CA
- automated CA (domain-based validation)
- ACME protocol in background
- easy TLS setup
- issued 100 million certs in June 2017

HTTPS Everywhere

- browser extension for Firefox & Chrome
- changes connections from HTTP to HTTPS
- rule-based
- manually maintained list

DANE

- DNS-based authentication of named entities
- replace PKI, ask DNS
- needs DNSSEC
- not used

TLS 1.3

Major differences

- static RSA removed
- forward secrecy everywhere
- CBC mode removed
- only AEAD (Authenticated Encryption with Associated Data)
- RC4, SHA1, MD5 removed
- compression removed
- renegotiation removed
- cipher suite changed
- Zero-RTT
- handshake state machine restructured
- fixed DHE groups

- session IDs + tickets - tickets + PSK
- downgrade protection
- full handshake signature