Network Fingerprinting

Theory, and Practice

Johanna Amann, Open Source Architect, Corelight

Fingerprinting?

Dictionary

Definitions from Oxford Languages · Learn more

fingerprint

/ˈfɪŋgəprɪnt/

noun

noun: **fingerprint**; plural noun: **fingerprints**

an impression or mark made on a surface by a person's fingertip, able to be used for identifying individuals from the unique pattern of whorls and lines on the fingertips.

- a distinctive identifying characteristic.

  "the faint chemical fingerprint of plastic explosives"

verb

verb: **fingerprint**; 3rd person present: **fingerprints**; past tense: **fingerprinted**; past participle: **fingerprinted**; gerund or present participle: **fingerprinting**

record the fingerprints of.

"I was booked, fingerprinted, and locked up for the night"

Source: google.com

FINGERPRINTING BY RANDOM POLYNOMIALS

by

Michael O. Rabin
Department of Mathematics
The Hebrew University of Jerusalem

and

Department of Computer Science
Harvard University

Center for Research in Computing Technology, Harvard University. Tech Report TR-CSE-03-01, 1981

Abstract

Randomly chosen irreducible polynomials $p(t) \in Z_2[t]$ are used to "fingerprint" bit-strings. This method is applied to produce a very simple real-time string matching algorithm and a procedure for securing files against unauthorized changes. The method is provably efficient and highly reliable for every input.

Center for Research in Computing Technology, Harvard University. Tech Report TR-CSE-03-01, 1981

Certificate Viewer: *.google.com

General | Details

Issued To

Common Name (CN)	*.google.com
Organisation (O)	<Not part of certificate>
Organisational Unit (OU)	<Not part of certificate>

Issued By

Common Name (CN)	WR2
Organisation (O)	Google Trust Services
Organisational Unit (OU)	<Not part of certificate>

Validity Period

Issued On	Wednesday, 3 December 2025 at 15:49:27
Expires On	Wednesday, 25 February 2026 at 15:49:26

SHA-256 Fingerprints

Certificate	0172d6c3fae57ef5ef15831feea5bf374c7802b4cdbf8def629f53b1b31ab6eb
Public key	837bccbc1989ecd909b703d5175606f10a7419f4885e06601420f530cc2424d0

FINGERPRINTING

Neal R. Wagner

Drexel University
Mathematical Sciences Department
Philadelphia, Pennsylvania 19104

Abstract. This paper presents a general discussion of the use of fingerprints, especially fingerprinted data. Fingerprinting is classified in four orthogonal ways, and some illustrative examples are given. The basis for a statistical analysis of altered fingerprints is presented, along with an example simulation. The possibility of more subtle fingerprints is discussed.

Proceedings of the 1983 IEEE Symposium on Security and Privacy

Network fingerprinting?

---[Phrack Magazine   Volume 8, Issue 54 Dec 25th, 1998, article 09 of 12

-------------------------[Remote OS detection via TCP/IP Stack FingerPrinting

--------[Fyodor <fyodor@dhp.com> (www.insecure.org) October 18, 1998

----[ABSTRACT

This paper discusses how to glean precious information about a host by querying its TCP/IP stack. I first present some of the "classical" methods of determining host OS which do not involve stack fingerprinting. Then I describe the current "state of the art" in stack fingerprinting tools. Next comes a description of many techniques for causing the remote host to leak information about itself. Finally I detail my (nmap) implementation of this, followed by a snapshot gained from nmap which discloses what OS is running on many popular Internet sites.

p0f - passive os fingerprinting tool

From: lcamtuf () TPI PL (Michal Zalewski)
Date: Sat, 10 Jun 2000 00:50:02 +0200

```
I'd like to announce beta release of p0f - passive OS fingerprinting
utility. I decided to publish it now, because I believe discussion will
help in process of debugging and developing next, stable version - and,
what's probably the most important - in collecting many different
fingerprints to include in database.
```

Remote physical device fingerprinting

Tadayoshi Kohno* Andre Broido† K.C. Claffy‡

May 25, 2005

Abstract

We introduce the area of *remote physical device fingerprinting*, or fingerprinting a physical device, as opposed to an operating system or class of devices, remotely, and without the fingerprinted device's known cooperation. We accomplish this goal by exploiting small, microscopic deviations in device hardware: clock skews. Our techniques do not require any modification to

IEEE Transactions on Dependable and Secure Computing (Volume: 2, Issue: 2, April-June 2005)

Fingerprint - definition

- A fingerprint means different things depending on context
  - Hash value uniquely identifying a file/datum
  - Value uniquely identifying a set of operating systems or programs
  - Value uniquely identifying a specific system
- Passive network fingerprinting - typically set of systems/programs

Motivation

- Threat hunting
  - Identifying commonalities between connections can be very helpful
  - Unique/unusual fingerprints can be worthwhile investigating
  - Mismatch detection in controlled environments
- Data enrichment

What to fingerprint?

Picture: Agence Olloweb @ Unsplash

TLS fingerprinting

Smarter Defending & Stealthier Attacking

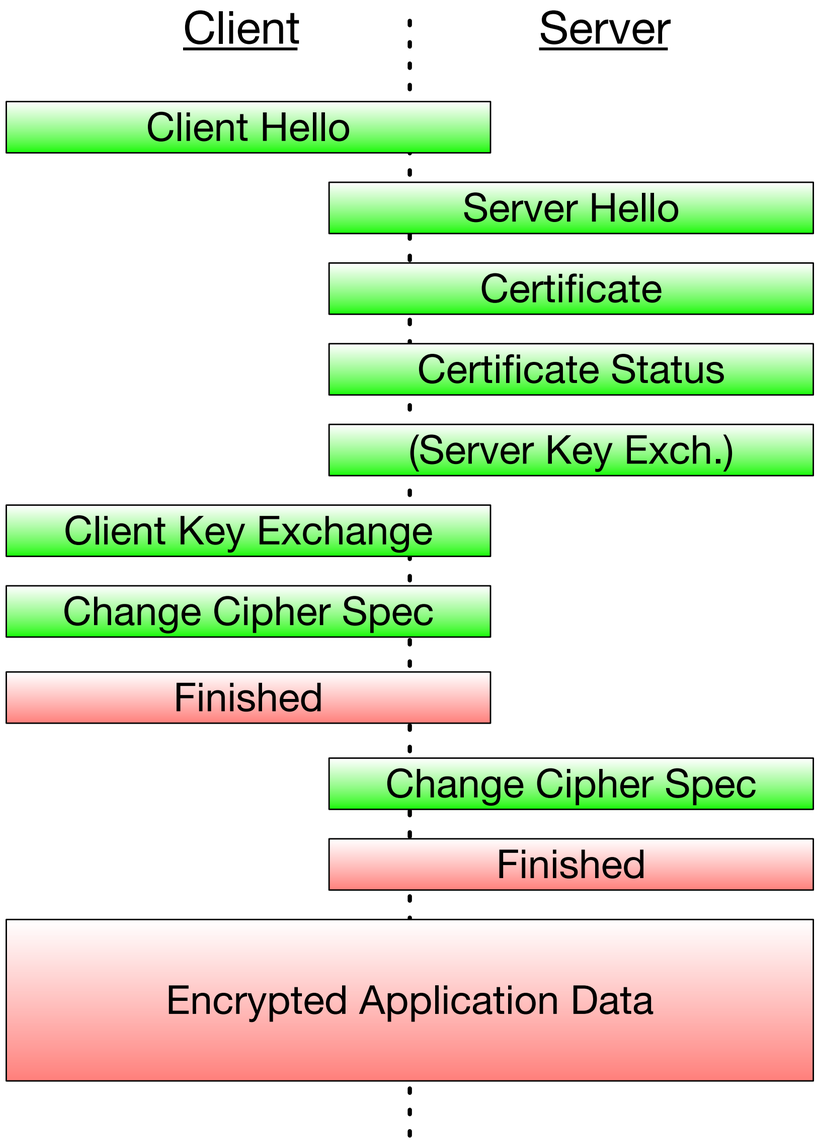
Posted on September 25, 2015

Background

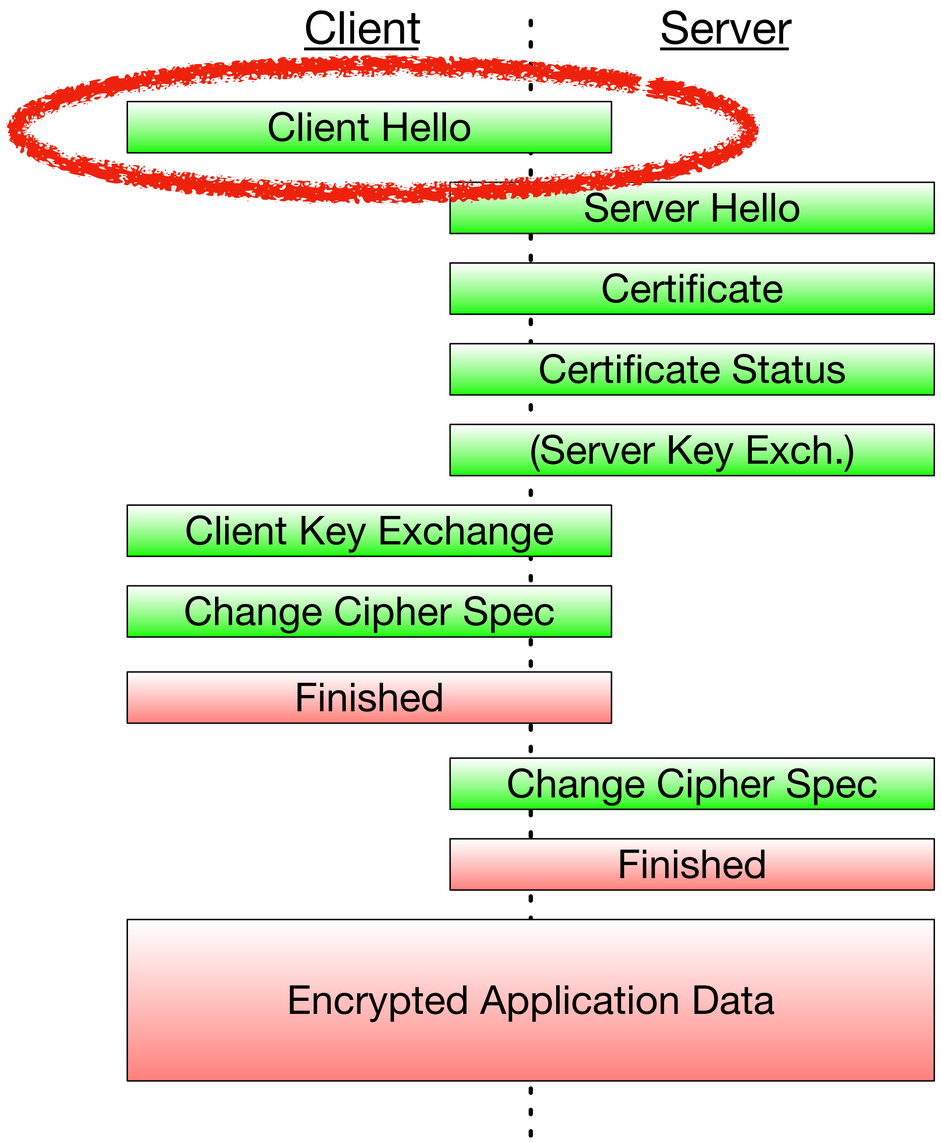
Transport Layer Security (TLS) provides security in the form of encryption to all manner of network connections from legitimate financial transactions, to private conversations, and malware calling home. The inability for an eavesdropper to analyze this encrypted traffic protects its users, whether they are legitimate or malicious. Those using TLS operate under the assumption that although an eavesdropper can easily observe the existence of their session, its source and destination IP addresses, that the content itself is secure and unreadable without access to cryptographic keying material at one or both ends of the connection. On the surface this holds true, barring any configuration flaws or exploitable vulnerabilities. However, using TLS Fingerprinting, it is easy to quickly and passively determine which client is being used, and then to apply this information from both the attacker and the defender perspectives.

Previously, I have been able to demonstrate that certain clients could be differentiated from other network traffic. Specifically, that meant discriminating SuperFish, PrivDog, and GeniusBox from mainstream browsers when making HTTPS connections, and generating IDS signatures based on these findings to assist network administrators in being able to identify problematic hosts without requiring access to either endpoint. I have now expanded this technique to improve the accuracy of the fingerprints; provide tools to enable others to create fingerprints; and tools that will enable use by others in their own environments.

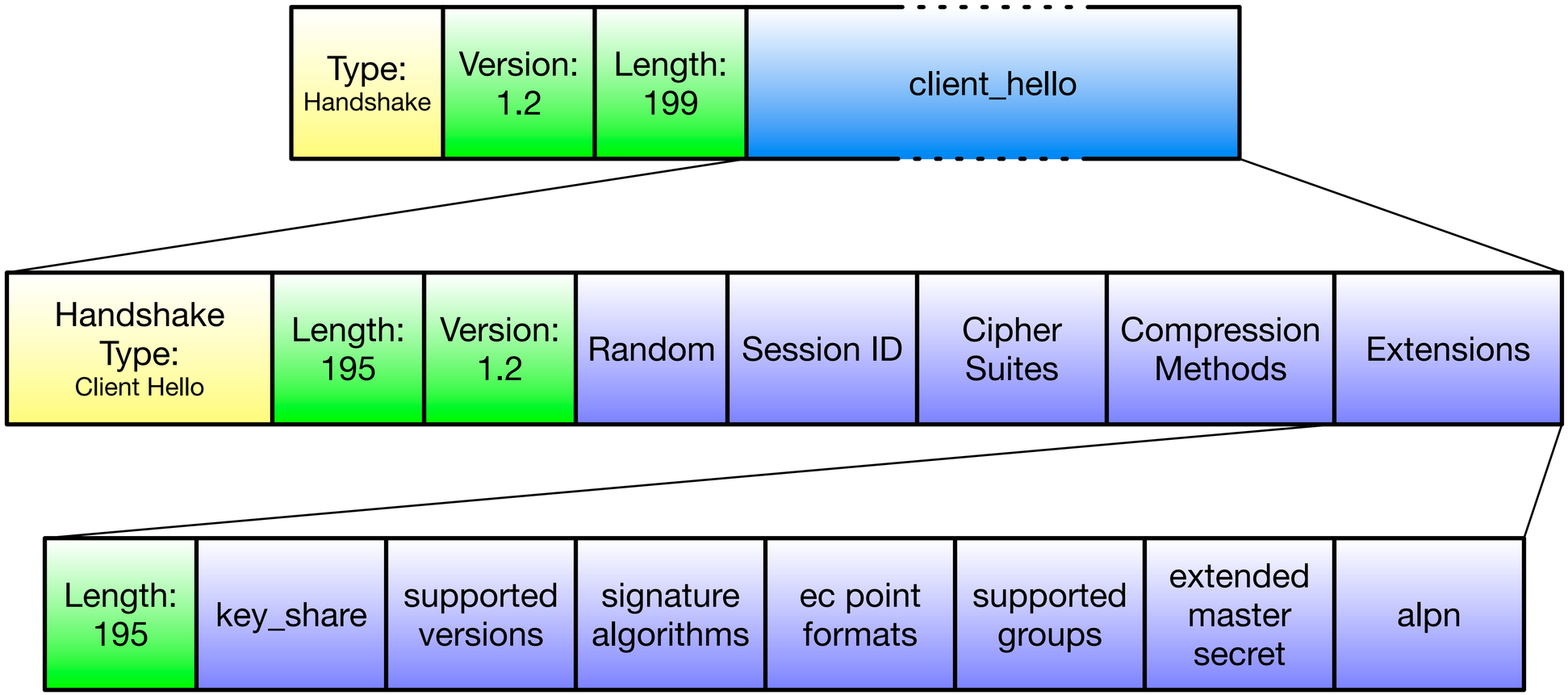
TLS 1.2


Client
Server
Client Hello
Server Hello
Certificate
Certificate Status
(Server Key Exch.)
Client Key Exchange
Change Cipher Spec
Finished
Change Cipher Spec
Finished
Encrypted Application Data

TLS 1.2


Client
Server
Client Hello
Server Hello
Certificate
Certificate Status
(Server Key Exch.)
Client Key Exchange
Change Cipher Spec
Finished
Change Cipher Spec
Finished
Encrypted Application Data

Client Hello



Fingerprinting - TLS

- Only need to look at first packet of connection, can be done statelessly
- Majority of Internet traffic
- Diverse set of client software
  - OpenSSL, GnuTLS, Schannel, NSS, go crypto/tls
  - Lots of ways to configure the protocol
- Payload is encrypted, so not directly usable
- Lots of unencrypted header data with lots of variability
  - Depending on used TLS version

Fingerprinting - TLS

atelessly

Number of TLS Flows:
67.5% (August 2020, large enterprise network)
78.8% (March 2022, large research facility)

Source: GGFAST: Automating Generation of Flexible Network Traffic Classifiers, Jilien Piet, Dubem Nwoji, Vern Paxson, ACM SIGCOMM '23: Proceedings of the ACM SIGCOMM 2023 Conference (September 2023)

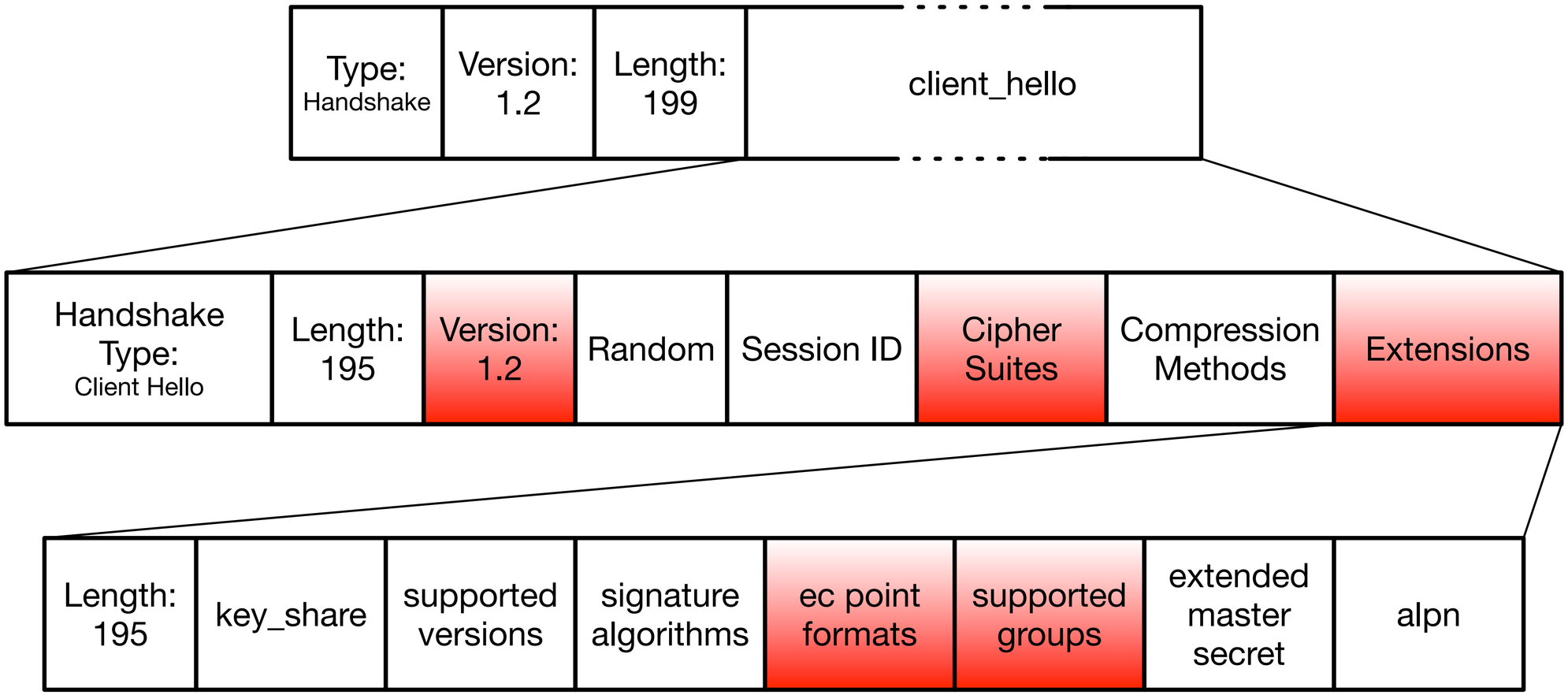
  - OpenSSL, GnuTLS, Schannel, NSS, go crypto/tls
  - Lots of ways to configure the protocol
- Payload is encrypted, so not directly usable
- Lots of unencrypted header data with lots of variability
  - Depending on used TLS version

Fingerprinting TLS - JA3

- TLS client (and later TLS server) fingerprint
- Creates short MD5-based hash from client hello
- Widespread
- BSD licensed
- Doesn’t deal with new randomisation approaches
- No longer maintained -> New version - JA4

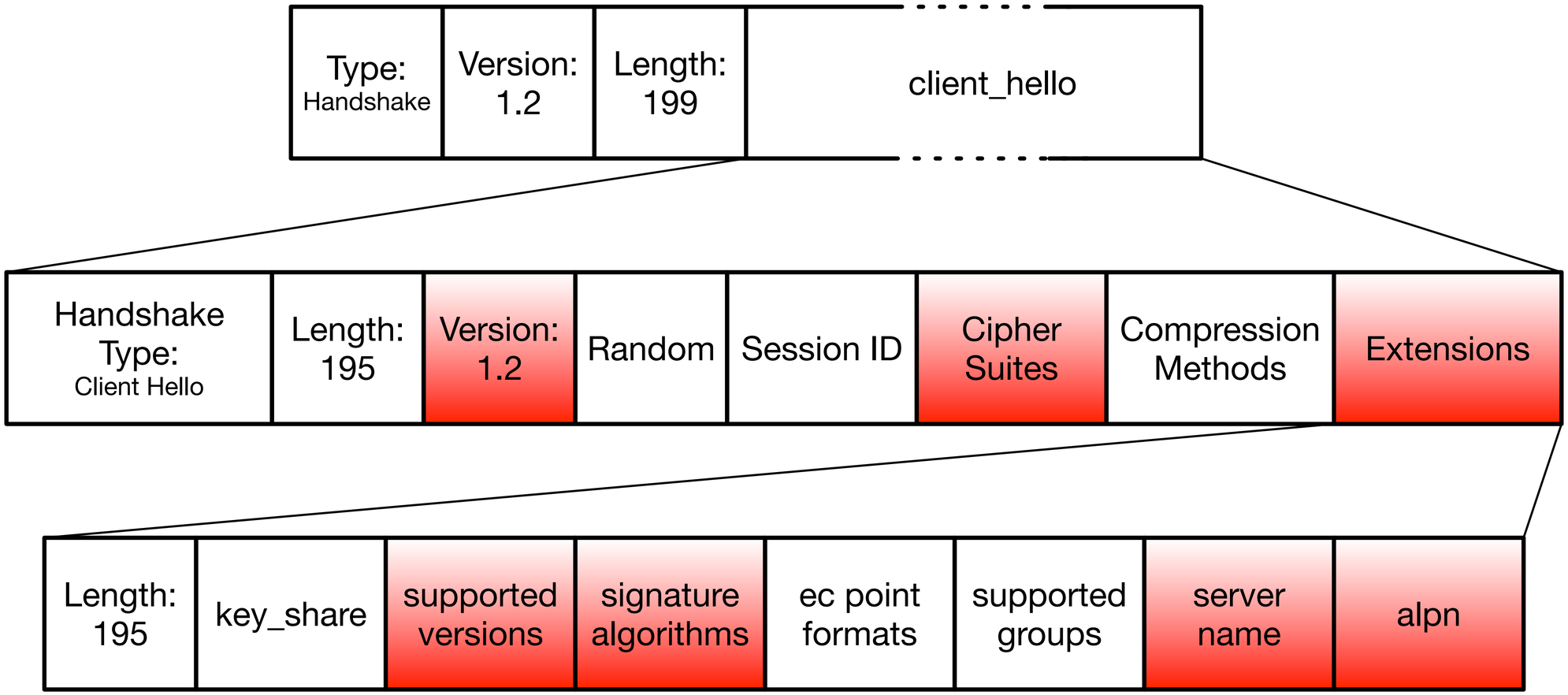
https://github.com/salesforce/ja3

JA3



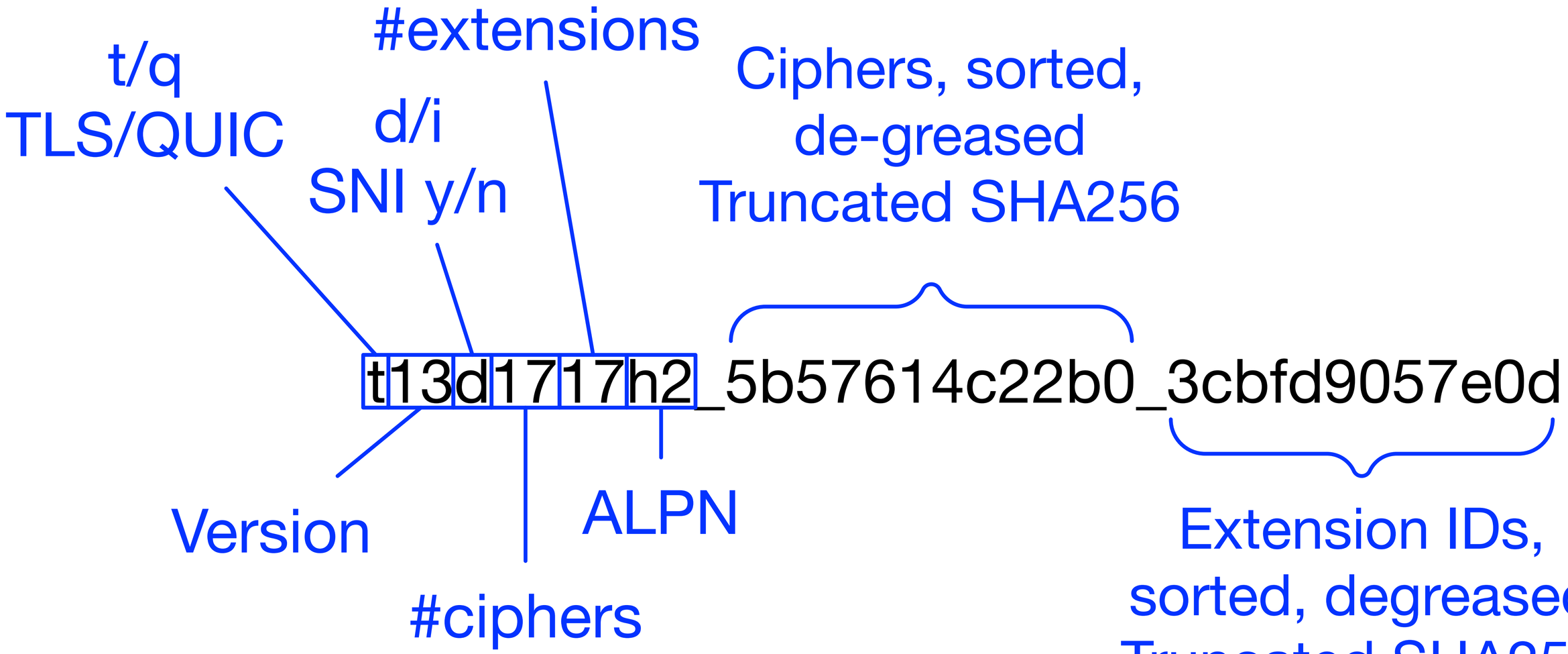
https://github.com/salesforce/ja3

JA4



https://github.com/FoxIO-LLC/ja4

Fingerprinting TLS - JA4



https://github.com/FoxIO-LLC/ja4

Fingerprinting - JA4+

- Suite of Fingerprints for different protocols
  - TLS Client Fingerprints (JA4)
  - TLS Server Fingerprints (JA4S)
  - HTTP Client Fingerprints (JA4HTTP)
  - Client to Server/Server to Client Latency (JA4L/JA4LS)
  - SSH (JA4SSH)
  - TCP Client/Server (JAT/JA4TS)
  - DHCP/DHCPv6 (JA4D/JA4DS)
- Proprietary license & patented. JA4 TLS Client Fingerprints BSD Licensed.

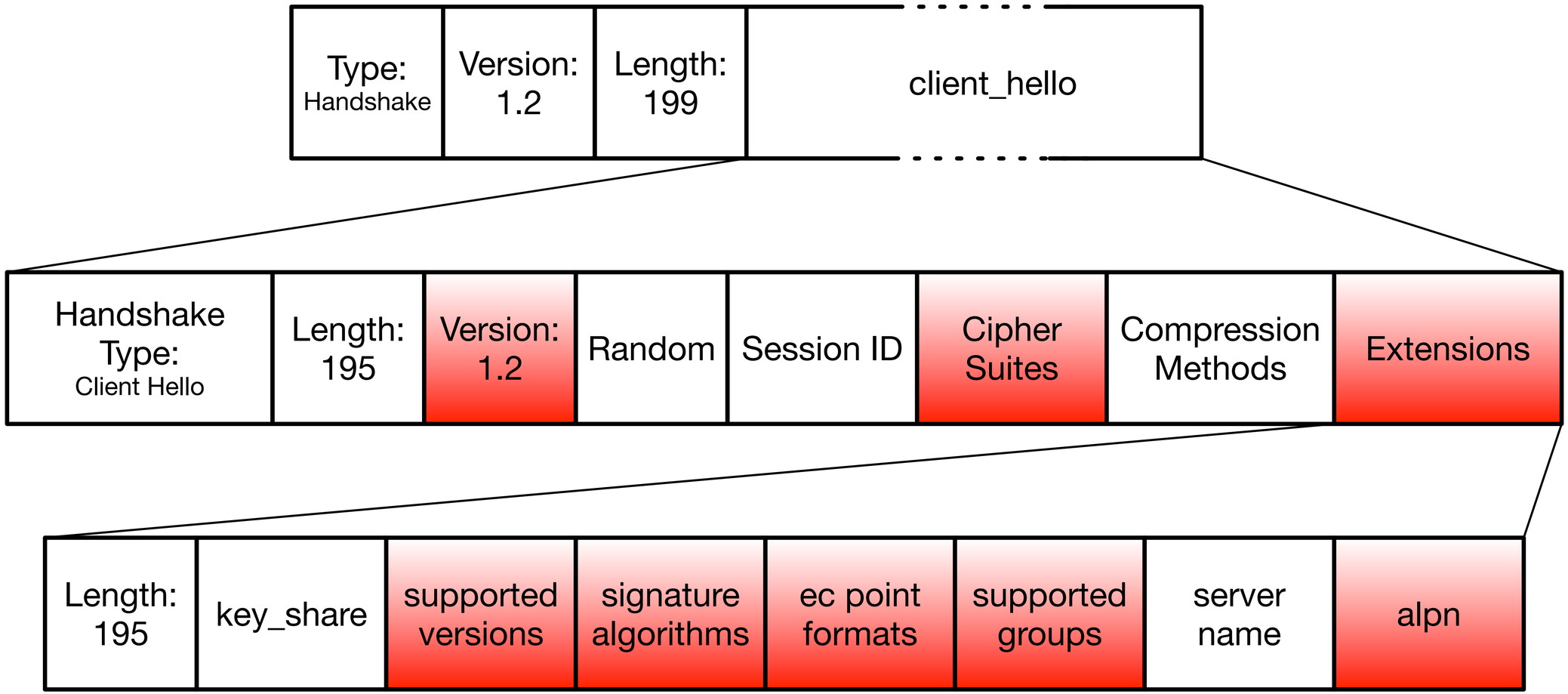
https://github.com/FoxIO-LLC/ja4

Mercury Network Protocol Fingerprinting

- Fingerprints for a range of network protocols
  - TCP
  - TLS/DTLS
  - QUIC
  - HTTP
  - OpenVPN
  - STUN
- BSD Licensed
- Technical specification, plus reference implementation

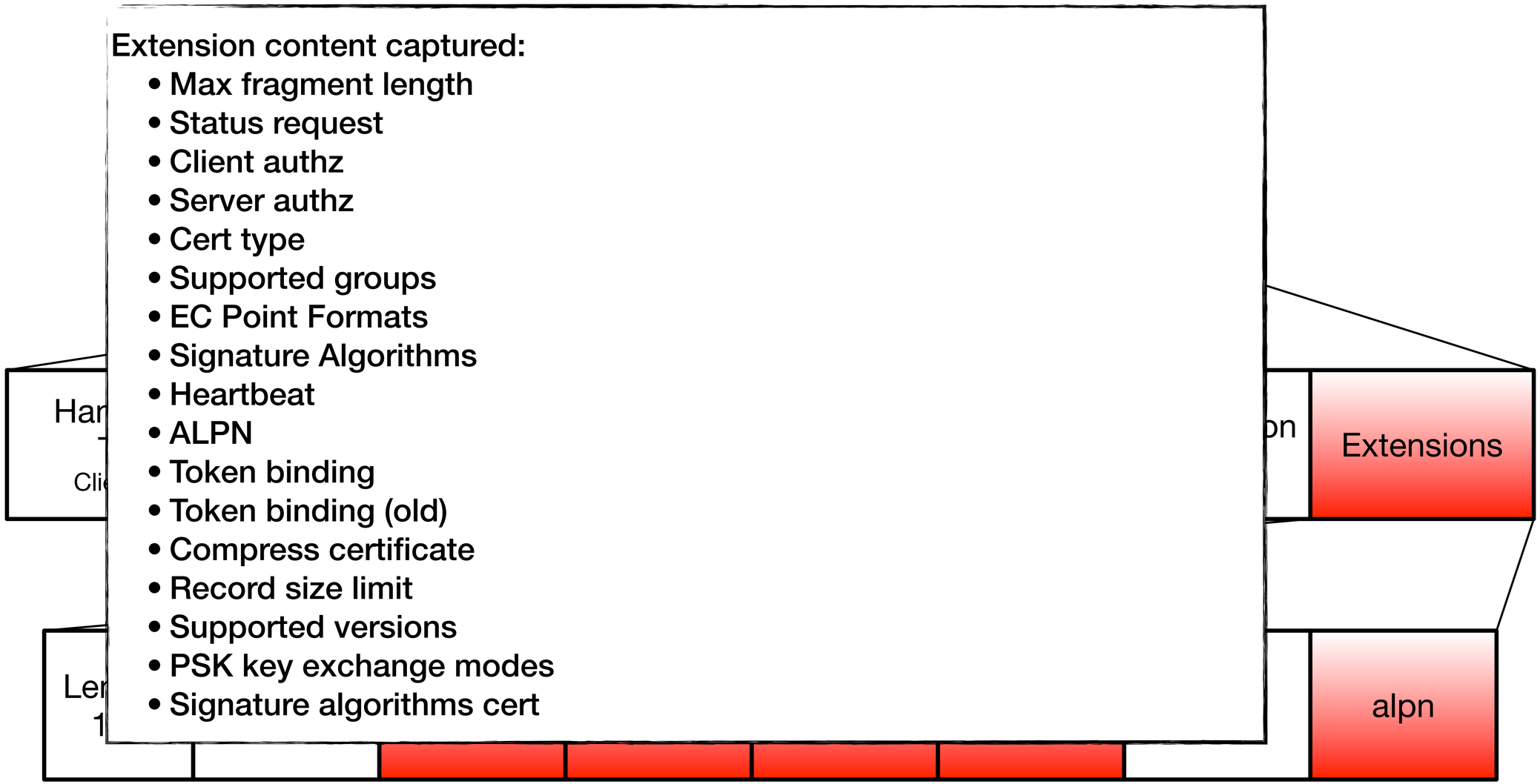
https://github.com/cisco/mercury

Mercury Network Protocol Fingerprinting



https://github.com/cisco/mercury

Mercury Network Protocol Fingerprinting



https://github.com/cisco/mercury

Mercury TLS Fingerprint

tls/2/(0303)(0a0a130213031301c02cc02bcca9c030c02fcca8c00ac009c014c013009d009c0035002fc008c012000a)
[(0000)(00050005010000000)(000a000e000cdada11ec001d001700180019)(000b00020100)
(000d00160014040308040401050308050805050108060601020 1)(0010000e000c02683208687474702f312e31)(0012)
(0017)(001b0003020001)(002b0007067a7a03040303)(002d00020101)(0033)(0a0a)(0a0a)(ff01)]

Mercury TLS Fingerprint

tls/2/(0303)(0a0a130213031301c02cc02bcca9c030c02fcca8c00ac009c014c013009d009c0035002fc008c012000a)
[(0000)(00050005010000000)(000a000e000cdada11ec001d00170018001 9)(000b00020100)
(000d00160014040308040401050308050805050108060601 0201)(0010000e000c02683208687474702f312e31)(0012)
(0017)(001b0003020001)(002b0007067a7a03040303)(002d00020101)(0033)(0a0a)(0a0a)(ff01)]

Mercury TLS Fingerprint

tls/fingerprint version/(tls version)(tls ciphersuites)[tls extensions, some with data]

Mercury Network Protocol Fingerprinting

- Specifically designed to allow several different kinds of matching:
  - Exact matches - use hash function on the fingerprint, compare
  - Prefix matches - for potentially truncated messages
  - Approximate matching - edit distance
  - Partial matching
- Protocols chance - supports versioning
- Data driven design

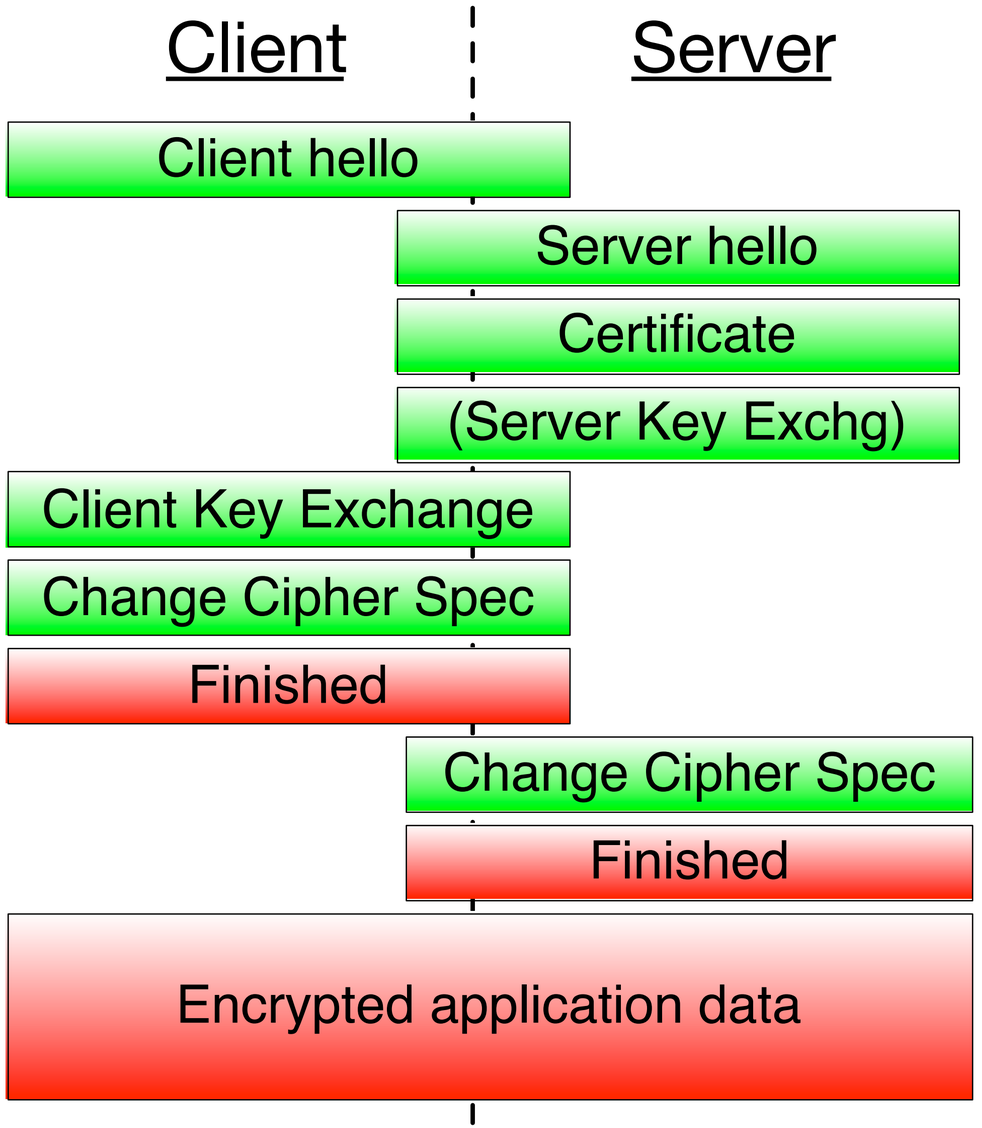
https://github.com/cisco/mercury

Other fingerprinting techniques

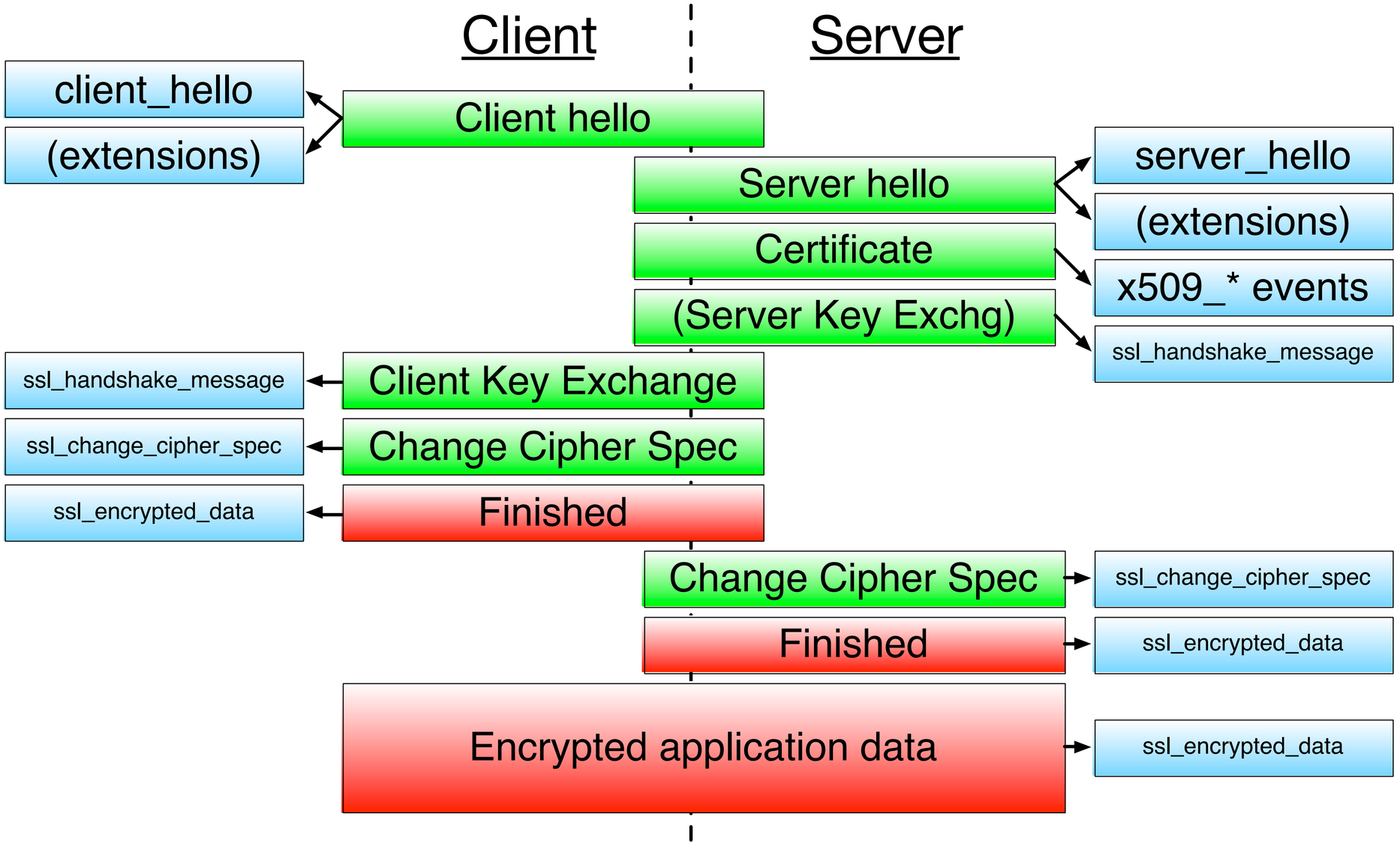
- Other protocols
  - DHCP Fingerprinting
  - HTTP/2 (Akamai)
  - DNS
- Sequencing/timing based analysis (e.g. GGFast)
- Packet Size Analysis

Implementation

TLS


Client
Server
Client hello
Server hello
Certificate
(Server Key Exchg)
Client Key Exchange
Change Cipher Spec
Finished
Change Cipher Spec
Finished
Encrypted application data

TLS


Client
Server
client_hello
(extensions)
Client hello
Server hello
server_hello
(extensions)
Certificate
x509_* events
(Server Key Exchg)
ssl_handshake_message
ssl_handshake_message
Client Key Exchange
ssl_change_cipher_spec
Change Cipher Spec
ssl_encrypted_data
Finished
Change Cipher Spec
ssl_change_cipher_spec
Finished
ssl_encrypted_data
Encrypted application data
ssl_encrypted_data

TLS Events

client_hello	ssl_stapled_ocsp	ssl_change_cipher_spec
server_hello	ssl_encrypted_data	x509_extension
ssl_session_ticket_handshake	ssl_dh_server_params	x509_ext_basic_constraints
ssl_established	ssl_change_cipher_spec	x509_ext_subject_alternative_name
x509_certificate	ssl_handshake_message	ssl_extension_elliptic_curves
ssl_extension	ssl_encrypted_data	ssl_extension_application_layer_protocol_negotiation
ssl_alert	ssl_extension_ec_point_formats	ssl_extension_server_name
ssl_server_curve	ssl_extension_signature_algorithm	x509_ocsp_ext_signed_certificate_timestamp
ssl_extension_supported_versions	ssl_extension_psk_key_exchange_modes	ssl_extension_signed_certificate_timestamp
ssl_stapled_ocsp	ssl_certificate_request	ssl_connection_flipped
ssl_probable_encrypted_handshake_message	ssl_plaintext_data	ssl_heartbeat
ssl_extension_connection_id	ssl_rsa_client_pms	ssl_dh_client_params
ssl_ecdh_client_params	ssl_server_signature	ssl_ecdh_server_params
ssl_extension_pre_shared_key_server_hello	ssl_extension_pre_shared_key_client_hello	ssl_extension_key_share

TLS Events

client_hello	ssl_stapled_ocsp	ssl_change_cipher_spec
server_hello	ssl_encrypted_data	x509_extension
ssl_session_ticket_handshake	ssl_dh_server_params	x509_ext_basic_constraints
ssl_established	ssl_change_cipher_spec	x509_ext_subject_alternative_name
x509_certificate	ssl_handshake_message	ssl_extension_elliptic_curves
ssl_extension	ssl_encrypted_data	ssl_extension_application_layer_protocol_negotiation
ssl_alert	ssl_extension_ec_point_formats	ssl_extension_server_name
ssl_server_curve	ssl_extension_signature_algorithm	x509_ocsp_ext_signed_certificate_timestamp
ssl_extension_supported_versions	ssl_extension_psk_key_exchange_modes	ssl_extension_signed_certificate_timestamp
ssl_stapled_ocsp	ssl_certificate_request	ssl_connection_flipped
ssl_probable_encrypted_handshake_message	ssl_plaintext_data	ssl_heartbeat
ssl_extension_connection_id	ssl_rsa_client_pms	ssl_dh_client_params
ssl_ecdh_client_params	ssl_server_signature	ssl_ecdh_server_params
ssl_extension_pre_shared_key_server_hello	ssl_extension_pre_shared_key_client_hello	ssl_extension_key_share

```
redef record SSL::Info += {
    mercury_tls_client_exts: vector of count &optional;
    mercury_tls_client_vals: vector of string &optional;
};

event ssl_extension(c: connection, is_client: bool, code: count, val: string) &priority=5
    {
    if ( ! is_client )
        return;

    if ( ! c$ssl?$mercury_tls_client_exts ) {
        c$ssl$mercury_tls_client_exts = vector();
        c$ssl$mercury_tls_client_vals = vector();
    }
    c$ssl$mercury_tls_client_exts[|c$ssl$mercury_tls_client_exts|] = code;
    c$ssl$mercury_tls_client_vals[|c$ssl$mercury_tls_client_vals|] = val;
    }
```

```
const TLS_GREASE: set[count] = {
    0x0a0a, 0x1a1a, 0x2a2a, 0x3a3a, 0x4a4a, 0x5a5a, 0x6a6a, 0x7a7a, 0x8a8a, 0x9a9a, 0xaaaa,
    0xbaba, 0xcaca, 0xdada, 0xeaea, 0xfafa
};

const TLS_EXT_FIXED: set[count] = {
    0x0001, 0x0005, 0x0007, 0x0008, 0x0009, 0x000a, 0x000b, 0x000d,
    0x000f, 0x0010, 0x0011, 0x0018, 0x001b, 0x001c, 0x002b, 0x002d,
    0x0032, 0x5500
};

event ssl_client_hello(c: connection, version: count, record_version: count, possible_ts: time, client_random: string, session_id: string, ciphers: index_vec, comp_methods: index_vec)
    {
    local unsorted_ciphers = degrease(ciphers);
    local tls_ext_string: string = "";
    if ( c$ssl?$mercury_tls_client_exts )
        {
        for ( i, ext in c$ssl$mercury_tls_client_exts )
            {
            if ( ext in TLS_EXT_FIXED )
                tls_ext_string += fmt("(%04x%04x%s)", ext, |c$ssl$mercury_tls_client_vals[i]|, bytestring_to_hexstr(c$ssl$mercury_tls_client_vals[i]));
            else
                tls_ext_string += fmt("(%04x)", degrease_single(ext));
            }
        }

    local tls_fp: string = fmt("tls/(%04x)(%s)(%s)", version, join_string_vec(unsorted_ciphers, ""), tls_ext_string);
    print tls_fp;
    }
```


zkg install corelight/zeek-mercury-npf

Result

```
$ zeek -C -i en0 packages
$ jq < ssl.log
{
  "ts": 1768317104.777162,
  "uid": "CinFi6oNIdj92KzO",
  "id.orig_h": “2a02:8010:…”,
  "id.orig_p": 53099,
  "id.resp_h": "2a00:1450:4009:c19::6c",
  "id.resp_p": 993,
  "version": "TLSv12",
  "cipher": "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
  "curve": "secp256r1",
  "server_name": "imap.gmail.com",
  "resumed": false,
  "established": true,
  "ssl_history": "CsxknGIi",
  "cert_chain_fps": [
    "4586f09f50b122d698c7ae55deb5ea8e06f56a0bb8844cfd6b9d084893ebc515",
    "e6fe22bf45e4f0d3b85c59e02c0f495418e1eb8d3210f788d48cd5e1cb547cd4",
    "3ee0278df71fa3c125c4cd487f01d774694e6fc57e0cd94c24efd769133918e5"
  ],
  "client_cert_chain_fps": [],
  "sni_matches_cert": true,
  "npf": "tls/(0303)(00ffc02cc02bc024c023c00ac009c008c030c02fc028c027c014c013c012009d009c003d003c0035002f000a)((0000)
(000a00080006001700180019)(000b00020100)(000d0012001004010201050106010403020305030603)(00050005010000000)(0012)(0017))"
}
```

Summary

- Fingerprints are a useful building block
- Valuable in nearly every scenario
- Many different applications - threat hunting, enrichment, policy enforcement
- The difference between fingerprints often is granularity
- Network protocols keep evolving
- High quality ground truth is not readily available