DANE and DNSSEC Authentication Chain Extension for TLS

Shumon Huque (with Melinda Shore, Richard Barnes, and Willem Toorop)

IETF 93, DANE Working Group
July 20th 2015, Prague, Czech Republic

https://tools.ietf.org/html/draft-shore-tls-dnssec-chain-extension-01

dnssec_chain

- A proposed new TLS extension.
- TLS server delivers a DANE record and the chain of DNSKEY & DS records needed to authenticate it.
- TLS client authenticates the record chain with a locally configured trust anchor (normally the root key).

wire format RRset

```
struct {
        opaque rrset<0..2^16-1>;
        opaque rrsig<0..2^16-1>;
} RRset
```

wire format RRsig record

```
RRset AuthenticationChain<0..2^16-1>;
```

Records are ordered starting from target DANE record going up to the trust anchor zone (normally the DNS root).

For the HTTPS site “www.example.com” where there are zone cuts at “com” & “example.com”:

DNSSEC chain will include the following order of RRsets (and corresponding RRsig records):

```
_443._tcp.www.example.com. TLSA
example.com. DNSKEY
example.com. DS
com. DNSKEY
com. DS
. DNSKEY
```

Rationale

- TLS client doesn’t need to perform the DANE related DNS queries itself:
  - avoids associated latency penalty.
  - works around middleboxes that might interfere with attempted DANE/DNSSEC queries.
- TLS client can authenticate the record itself without needing access to a validating resolver to which it has a secure connection.

Rationale

- Another possible use case:
  - Link layer authentication, e.g. 802.1X and EAP methods that employ TLS: EAP-TLS, EAP-TTLS, TEAP, etc.
  - TLS client cannot do any DNS queries by design in this case.

Server side

- Build the DNSSEC authentication chain.
- Return the chain in the dnssec_chain extension of ServerHello when the client asks for it.
- Cache and reuse it across multiple connections.
- Periodically rebuild the chain as TTLs and signature validity periods require.

Client side

- Send dnssec_chain extension in ClientHello.
- Obtain dnssec chain from server and authenticate it with locally configured trust anchor.
- Use DANE record to authenticate the server’s certificate.
- Perform trust anchor maintenance (RFC 5011), or obtain this via an external service.

Future work

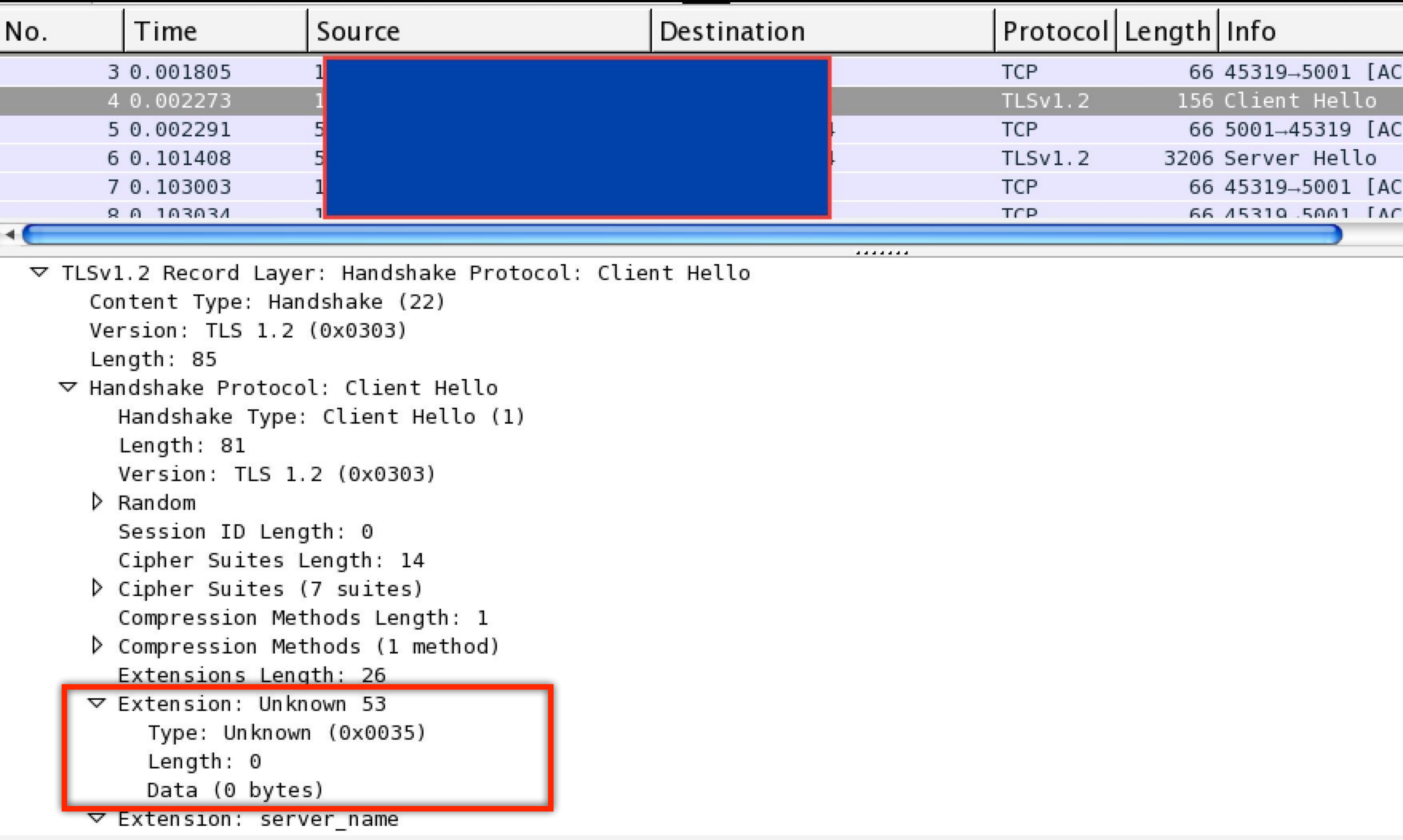
- Dealing properly with CNAME, DNAME, and wildcards.

Under consideration ..

- Client can cache DNS records.
- Client sends list of unexpired cached records it possesses to server.
- Server delivers shorter chain with those records omitted.

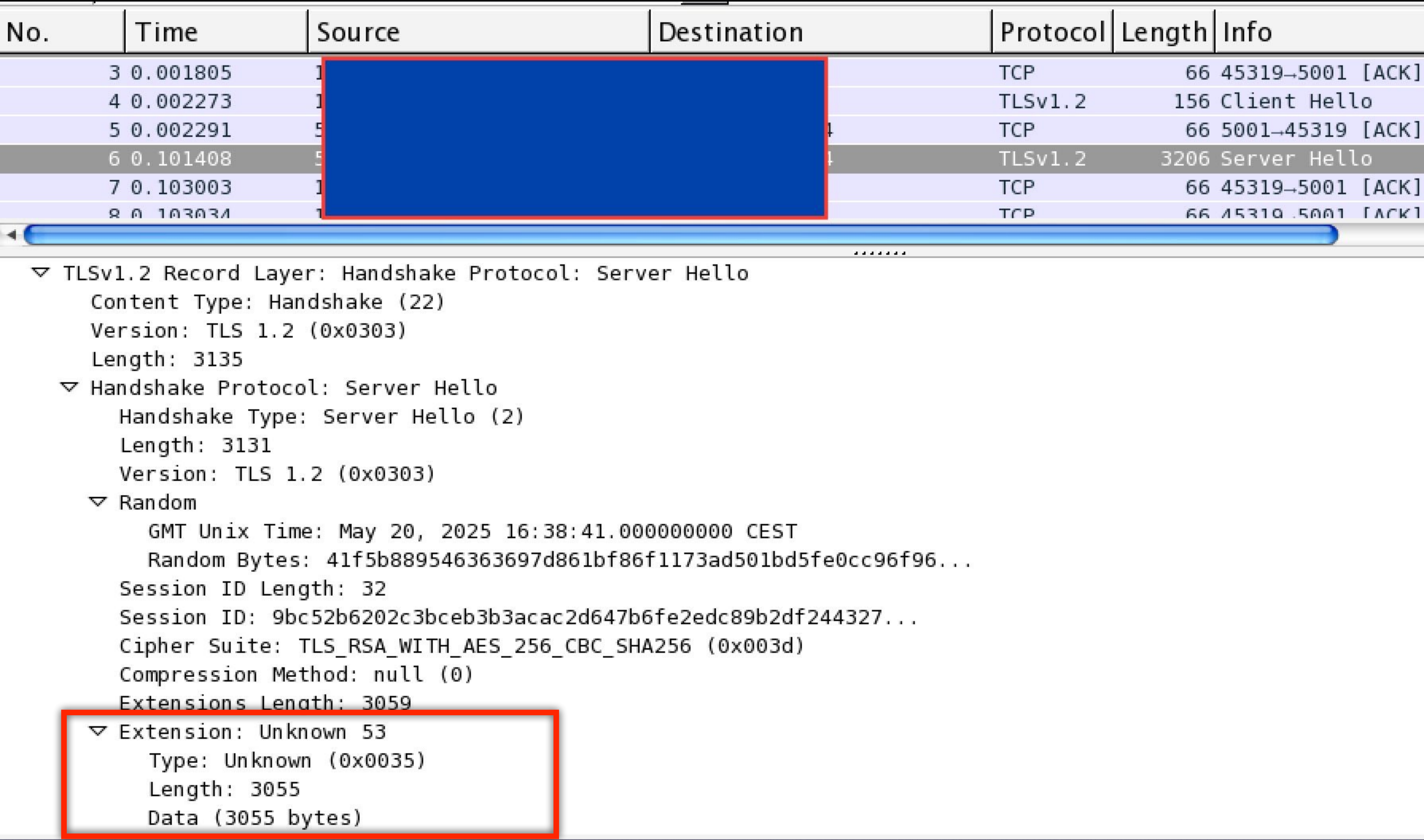
Prototypes?

- Prototype code being developed.



```
No.   Time       Source  Destination  Protocol  Length  Info
    3 0.001805   [redacted]           TCP           66  45319→5001 [AC
    4 0.002273                        TLSv1.2      156  Client Hello
    5 0.002291                        TCP           66  5001→45319 [AC
    6 0.101408                        TLSv1.2     3206  Server Hello
    7 0.103003                        TCP           66  45319→5001 [AC
    8 0.103034                        TCP           66  45319→5001 [AC
```

```
▽ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 85
  ▽ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 81
      Version: TLS 1.2 (0x0303)
    ▷ Random
      Session ID Length: 0
      Cipher Suites Length: 14
    ▷ Cipher Suites (7 suites)
      Compression Methods Length: 1
    ▷ Compression Methods (1 method)
      Extensions Length: 26
    ▽ Extension: Unknown 53
        Type: Unknown (0x0035)
        Length: 0
        Data (0 bytes)
    ▽ Extension: server_name
```



```
No.    Time       Source       Destination       Protocol  Length  Info
     3 0.001805   [redacted]                     TCP           66  45319→5001 [ACK]
     4 0.002273                                  TLSv1.2      156  Client Hello
     5 0.002291                                  TCP           66  5001→45319 [ACK]
     6 0.101408                                  TLSv1.2     3206  Server Hello
     7 0.103003                                  TCP           66  45319→5001 [ACK]
     8 0.103034                                  TCP           66  45319→5001 [ACK]
```

```
▽ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 3135
  ▽ Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 3131
      Version: TLS 1.2 (0x0303)
    ▽ Random
        GMT Unix Time: May 20, 2025 16:38:41.000000000 CEST
        Random Bytes: 41f5b889546363697d861bf86f1173ad501bd5fe0cc96f96...
      Session ID Length: 32
      Session ID: 9bc52b6202c3bceb3b3acac2d647b6fe2edc89b2df244327...
      Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
      Compression Method: null (0)
      Extensions Length: 3059
    ▽ Extension: Unknown 53
        Type: Unknown (0x0035)
        Length: 3055
        Data (3055 bytes)
```

More details

https://tools.ietf.org/html/draft-shore-tls-dnssec-chain-extension-01