Using EAP-TLS with TLS 1.3

draft-ietf-emu-eap-tls13-00

EMU IETF 102, Montreal, July 2018, John Mattsson

emu by Jon Bunting https://www.flickr.com/photos/84744710@N06/14766013011

DRAFT-IETF-EAP-TLS13-00

- Now a working group document. Changes since draft-mattsson-eap-tls13-02:
  - Editorial changes
  - Rewritten text on resumption:

    ”It is RECOMMENDED that the EAP server accept resumption as long as the ticket is valid. However, the server MAY choose to require a full authentication.”

  - Updated the TLS exporter labels to follow RFC 5705 and added IANA considerations:

```
Key_Material = TLS-Exporter("EXPORTER_EAP_TLS_Key_Material", "", 128)
IV           = TLS-Exporter("EXPORTER_EAP_TLS_IV", "", 64)
Session-Id   = TLS-Exporter("EXPORTER_EAP_TLS_Session-Id", "", 64)
```

- Implementation and comments by Jouni Malinen

NEWSESSIONTICKET ISSUES

EAP Server not supporting resumption

```
 EAP Peer                                                EAP Server

                                                     EAP-Request/
                            <--------                    Identity
 EAP-Response/
 Identity (MyID)            -------->
                                                     EAP-Request/
                                                 EAP-Type=EAP-TLS
                            <--------                 (TLS Start)
 EAP-Response/
 EAP-Type=EAP-TLS
(TLS ClientHello)           -------->
                                                     EAP-Request/
                                                 EAP-Type=EAP-TLS
                                                (TLS ServerHello,
                                         TLS EncryptedExtensions,
                                          TLS CertificateRequest,
                                                 TLS Certificate,
                                           TLS CertificateVerify,
                            <--------               TLS Finished)
 EAP-Response/
 EAP-Type=EAP-TLS
(TLS Certificate,
 TLS CertificateVerify,
 TLS Finished)              -------->
                            <--------                 EAP-Success
```

?

EAP Server supporting resumption

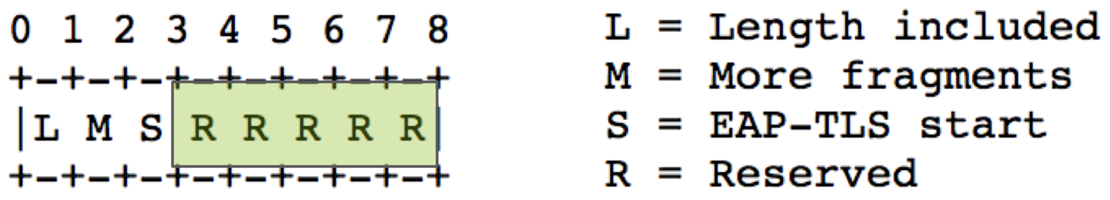
```
 EAP Peer                                                EAP Server

                                                     EAP-Request/
                            <--------                    Identity
 EAP-Response/
 Identity (MyID)            -------->
                                                     EAP-Request/
                                                 EAP-Type=EAP-TLS
                            <--------                 (TLS Start)
 EAP-Response/
 EAP-Type=EAP-TLS
(TLS ClientHello)           -------->
                                                     EAP-Request/
                                                 EAP-Type=EAP-TLS
                                                (TLS ServerHello,
                                         TLS EncryptedExtensions,
                                          TLS CertificateRequest,
                                                 TLS Certificate,
                                           TLS CertificateVerify,
                            <--------               TLS Finished)
 EAP-Response/
 EAP-Type=EAP-TLS
(TLS Certificate,
 TLS CertificateVerify,
 TLS Finished)              -------->
                                                     EAP-Request/
                                                 EAP-Type=EAP-TLS
                            <--------       (TLS NewSessionTicket)
 EAP-Response/
 EAP-Type=EAP-TLS           -------->
                            <--------                 EAP-Success
```

NEWSESSIONTICKET ISSUES

- EAP Peer does not know whether the NewSessionTicket will be delivered after ClientFinished.
  - The next message in the sequence could be either continuation of EAP-TLS method or EAP-Success making the RFC 4137 state machine dependent on TLS version
  - TLS 1.0, 1.1, 1.2: methodState=DONE, decision=UNCOND_SUCC
  - TLS 1.3: methodState=MAY_CONT, decision=COND_SUCC
- Jouni states that this is *"a bit inconvinient"* and asks if there are ways to avoid the uncertainty and latency.
- **Is the uncertainty and latency something that should be addressed?**
- An TLS 1.3 server could theoretically send several NewSessionTicket and other Post-Handshake Messages (Section 4.6 in TLS 1.3) after the main handshake.
- **Should EAP-TLS supports all Post-Handshake Messages or only a single NewSessionTicket?**

NEWSESSIONTICKET UNCERTAINTY

- The ’Flags’ byte sent in EAP-TLS Request and Response packets could potentially be used to reduce uncertainty. The Server could set some bits in the EAP-Request containing it’s Finished message.



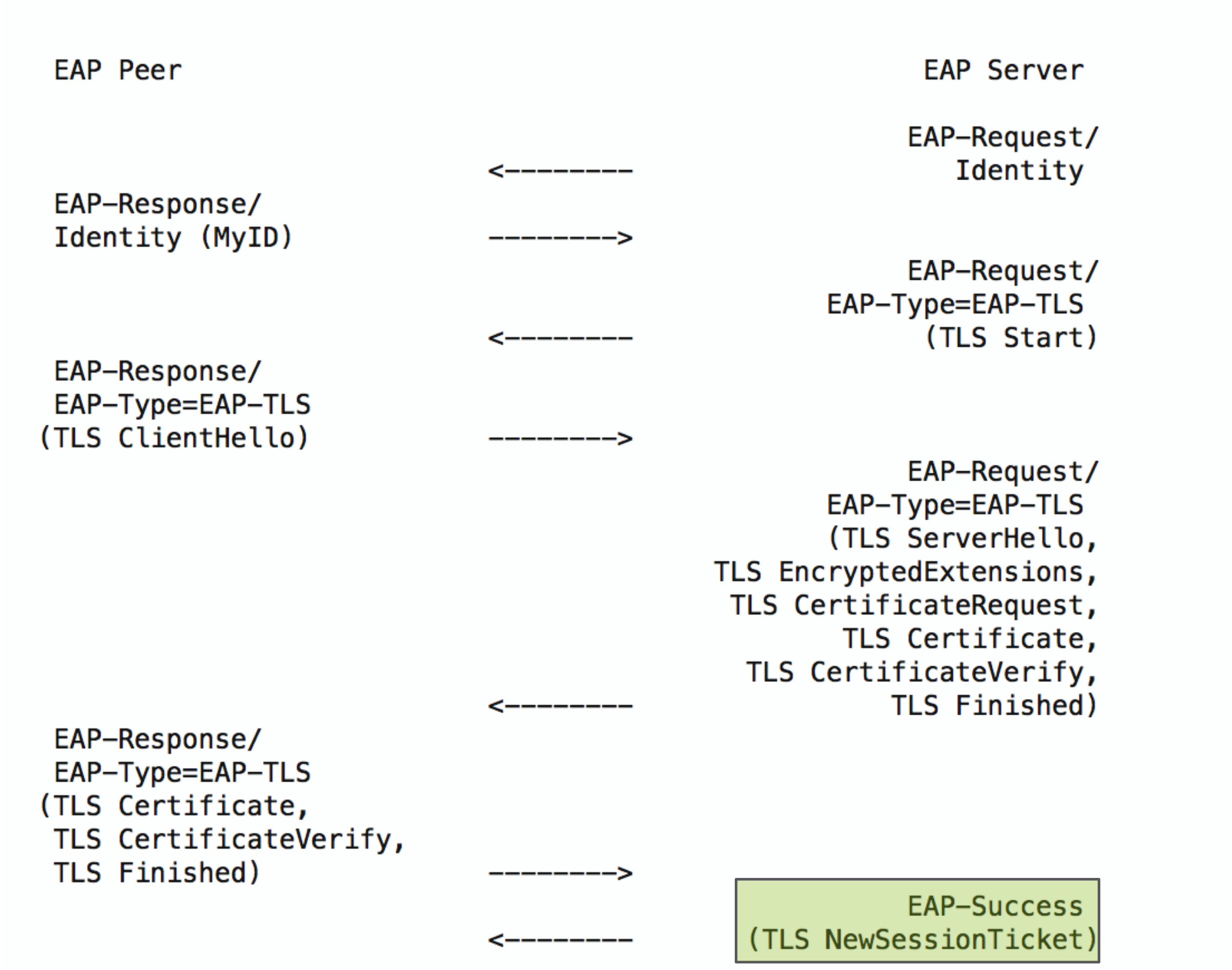
```
0 1 2 3 4 5 6 7 8       L = Length included
+-+-+-+-+-+-+-+-+       M = More fragments
|L M S R R R R R|       S = EAP-TLS start
+-+-+-+-+-+-+-+-+       R = Reserved
```

- **Does the TLS server know whether it will send more Post-Handshake Messages (like NewSessionTicket) before receiving the Finished message from the TLS client?**
- **How much information does the EAP-TLS layer gets from the TLS layer?**

NEWSESSIONTICKET LATENCY

- Jouni suggests piggybacking NewSessionTicket on top of the EAP-Success message.
- Would remove both uncertainty and latency.
- Would require an update of RFC 3748.
- **Opinions?**

Server supporting resumption



```
 EAP Peer                                          EAP Server

                                                EAP-Request/
                            <--------               Identity
 EAP-Response/
 Identity (MyID)            -------->
                                                EAP-Request/
                                             EAP-Type=EAP-TLS
                            <--------             (TLS Start)
 EAP-Response/
 EAP-Type=EAP-TLS
(TLS ClientHello)           -------->
                                                EAP-Request/
                                             EAP-Type=EAP-TLS
                                            (TLS ServerHello,
                                     TLS EncryptedExtensions,
                                      TLS CertificateRequest,
                                             TLS Certificate,
                                       TLS CertificateVerify,
                            <--------           TLS Finished)
 EAP-Response/
 EAP-Type=EAP-TLS
(TLS Certificate,
 TLS CertificateVerify,
 TLS Finished)              -------->
                                                 EAP-Success
                            <--------   (TLS NewSessionTicket)
```

KEY DERIVATION

- The key derivation has been causing interoperability problems for EAP-TLS in the past.
- RFC 5216:

```
Key_Material = TLS-PRF-128(master_secret, "client EAP encryption",
                           client.random || server.random)
IV           = TLS-PRF-64("", "client EAP encryption",
Session-Id   = 0x0D || client.random || server.random
```

- draft-ietf-eap-tls13:

```
Key_Material  = TLS-Exporter("EXPORTER_EAP_TLS_Key_Material", "", 128)
IV            = TLS-Exporter("EXPORTER_EAP_TLS_IV", "", 64)
Session-Id    = TLS-Exporter("EXPORTER_EAP_TLS_Session-Id", "", 64)
```

- The Key_Material derivation in RFC 5216 is compliant with the TLS-exporter interface (RFC 5705) Key_Material = TLS-Exporter("client EAP encryption", null, 128). The IV derivation is not.
- The Session-ID definition requires that the EAP Peer and EAP Server to read 32 bytes at TLS_Data[6] to get the random numbers.
- TLS-exporter change got support on list, Jouni states that the dependency on TLS version is *“a bit inconvinient”*
- **What is the best tradeoff between implementation convenient, what the API is supposed to be between TLS and EAP-TLS, and security?**
- **We should document the interface between EAP-TLS and TLS.**

WANTED

FEEDBACK

REVIEWS

IMPLEMENTATIONS

INTEROP