TLS
Internet-Draft
Intended status: Informational
Expires: January 10, 2019

M. Thomson
Mozilla

July 09, 2018

Example Handshake Traces for TLS 1.3
draft-ietf-tls-tls13-vectors-06

Abstract

Examples of TLS 1.3 handshakes are shown. Private keys and inputs are provided so that these handshakes might be reproduced. Intermediate values, including secrets, traffic keys and IVs are shown so that implementations might be checked incrementally against these values.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 10, 2019.

Table of Contents

1. Introduction 2
2. Private Keys 2
3. Simple 1-RTT Handshake 3
4. Resumed 0-RTT Handshake 15
5. HelloRetryRequest 26
6. Client Authentication 38
7. Compatibility Mode 49
8. Security Considerations 60
9. IANA Considerations 60
10. References 60
10.1. Normative References 60
10.2. Informative References 60
Appendix A. Acknowledgements 61
Author's Address 61

1. Introduction

TLS 1.3 [TLS13] defines a new key schedule and a number of new cryptographic operations. This document includes sample handshakes that show all intermediate values. This allows an implementation to be verified incrementally, examining inputs and outputs of each cryptographic computation independently.

A private key is included with the traces so that implementations can be checked by importing these values and verifying that the same outputs are produced.

Note: Invocations of HMAC-based Extract-and-Expand Key Derivation Function (HKDF) [RFC5869] are not labelled, but can be identified through the use the labels used by HKDF.

2. Private Keys

Ephemeral private keys are shown as they are generated in the traces.

The server in most examples uses an RSA certificate with a private key of:

modulus (public): b4 bb 49 8f 82 79 30 3d 98 08 36 39 9b 36 c6 98 8c
0c 68 de 55 e1 bd b8 26 d3 90 1a 24 61 ea fd 2d e4 9a 91 d0 15 ab
bc 9a 95 13 7a ce 6c 1a f1 9e aa 6a f9 8c 7c ed 43 12 09 98 e1 87

```
   a8 0e e0 cc b0 52 4b 1b 01 8c 3e 0b 63 26 4d 44 9a 6d 38 e2 2a 5f
   da 43 08 46 74 80 30 53 0e f0 46 1c 8c a9 d9 ef bf ae 8e a6 d1 d0
   3e 2b d1 93 ef f0 ab 9a 80 02 c4 74 28 a6 d3 5a 8d 88 d7 9f 7f 1e
   3f
```



```
public exponent:  01 00 01

private exponent:  04 de a7 05 d4 3a 6e a7 20 9d d8 07 21 11 a8 3c 81
   e3 22 a5 92 78 b3 34 80 64 1e af 7c 0a 69 85 b8 e3 1c 44 f6 de 62
   e1 b4 c2 30 9f 61 26 e7 7b 7c 41 e9 23 31 4b bf a3 88 13 05 dc 12
   17 f1 6c 81 9c e5 38 e9 22 f3 69 82 8d 0e 57 19 5d 8c 84 88 46 02
   07 b2 fa a7 26 bc f7 08 bb d7 db 7f 67 9f 89 34 92 fc 2a 62 2e 08
   97 0a ac 44 1c e4 e0 c3 08 8d f2 5a e6 79 23 3d f8 a3 bd a2 ff 99
   41

prime1:  e4 35 fb 7c c8 37 37 75 6d ac ea 96 ab 7f 59 a2 cc 10 69 db
   7d eb 19 0e 17 e3 3a 53 2b 27 3f 30 a3 27 aa 0a aa bc 58 cd 67 46
   6a f9 84 5f ad c6 75 fe 09 4a f9 2c 4b d1 f2 c1 bc 33 dd 2e 05 15

prime2:  ca bd 3b c0 e0 43 86 64 c8 d4 cc 9f 99 97 7a 94 d9 bb fe ad
   8e 43 87 0a ba e3 f7 eb 8b 4e 0e ee 8a f1 d9 b4 71 9b a6 19 6c f2
   cb ba ee eb f8 b3 49 0a fe 9e 9f fa 74 a8 8a a5 1f c6 45 62 93 03

exponent1:  3f 57 34 5c 27 fe 1b 68 7e 6e 76 16 27 b7 8b 1b 82 64 33
   dd 76 0f a0 be a6 a6 ac f3 94 90 aa 1b 47 cd a4 86 9d 68 f5 84 dd
   5b 50 29 bd 32 09 3b 82 58 66 1f e7 15 02 5e 5d 70 a4 5a 08 d3 d3
   19

exponent2:  18 3d a0 13 63 bd 2f 28 85 ca cb dc 99 64 bf 47 64 f1 51
   76 36 f8 64 01 28 6f 71 89 3c 52 cc fe 40 a6 c2 3d 0d 08 6b 47 c6
   fb 10 d8 fd 10 41 e0 4d ef 7e 9a 40 ce 95 7c 41 77 94 e1 04 12 d1
   39

coefficient:  83 9c a9 a0 85 e4 28 6b 2c 90 e4 66 99 7a 2c 68 1f 21
   33 9a a3 47 78 14 e4 de c1 18 33 05 0e d5 0d d1 3c c0 38 04 8a 43
   c5 9b 2a cc 41 68 89 c0 37 66 5f e5 af a6 05 96 9f 8c 01 df a5 ca
   96 9d
```

3. Simple 1-RTT Handshake

In this example, the simplest possible handshake is completed. The

server is authenticated, but the client remains anonymous. After connecting, a few application data octets are exchanged. The server sends a session ticket that permits the use of 0-RTT data in any resumed session.

{client} create an ephemeral x25519 key pair:

private key (32 octets): 01 61 d7 bf 4b a0 6c 35 68 f1 09 54 f0 f1 ca 08 74 60 54 9c dc 7b fe b2 77 6b 46 04 d8 2f aa c2

public key (32 octets): b0 f5 01 9f b0 f1 e5 37 6b 8b 1d fb 90 5f 1d 91 51 61 ba c3 77 07 da d8 90 7b d7 1b 98 07 b3 45

{client} send a ClientHello handshake message

{client} send handshake record:

payload (196 octets): 01 00 00 c0 03 03 d4 b9 50 3c 5e 95 c9 ee cc 99 ce 63 76 cc ad 4d cc 06 d7 c8 f1 fa 44 b0 d9 56 00 e9 a0 58 6c 67 00 00 06 13 01 13 03 13 02 01 00 00 91 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 14 00 12 00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 23 00 00 00 33 00 26 00 24 00 1d 00 20 b0 f5 01 9f b0 f1 e5 37 6b 8b 1d fb 90 5f 1d 91 51 61 ba c3 77 07 da d8 90 7b d7 1b 98 07 b3 45 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02 05 02 06 02 02 02 00 2d 00 02 01 01 00 1c 00 02 40 01

ciphertext (201 octets): 16 03 01 00 c4 01 00 00 c0 03 03 d4 b9 50 3c 5e 95 c9 ee cc 99 ce 63 76 cc ad 4d cc 06 d7 c8 f1 fa 44 b0 d9 56 00 e9 a0 58 6c 67 00 00 06 13 01 13 03 13 02 01 00 00 91 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 14 00 12 00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 23 00 00 00 33 00 26 00 24 00 1d 00 20 b0 f5 01 9f b0 f1 e5 37 6b 8b 1d fb 90 5f 1d 91 51 61 ba c3 77 07 da d8 90 7b d7 1b 98 07 b3 45 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02 05 02 06 02 02 02 00 2d 00 02 01 01 00 1c 00 02 40 01

{server} extract secret "early":

salt: (absent)

```
   IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
      e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

{server}  create an ephemeral x25519 key pair:

   private key (32 octets):  e2 36 b9 50 e1 aa 9b af af ed c6 d1 c9
      31 18 67 fd 56 91 d2 c1 5e 05 3b 5a b0 85 f7 3f 75 a8 6a

   public key (32 octets):  9d 3c 94 0d 89 69 0b 84 d0 8a 60 99 3c 14
      4e ca 68 4d 10 81 28 7c 83 4d 53 11 bc f3 2b b9 da 1a

{server}  send a ServerHello handshake message

{server}  derive secret for handshake "tls13 derived":
```



```
   PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
      10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
      97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{server}  extract secret "handshake":

   salt (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
      16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

   IKM (32 octets):  81 51 d1 46 4c 1b 55 53 36 23 b9 c2 24 6a 6a 0e
      6e 7e 18 50 63 e1 4a fd af f0 b6 e1 c6 1a 86 42

   secret (32 octets):  5b 4f 96 5d f0 3c 68 2c 46 e6 ee 86 c3 11 63
```

```
      66 15 a1 d2 bb b2 43 45 c2 52 05 95 3c 87 9e 8d 06

{server}  derive secret "tls13 c hs traffic":

   PRK (32 octets):  5b 4f 96 5d f0 3c 68 2c 46 e6 ee 86 c3 11 63 66
      15 a1 d2 bb b2 43 45 c2 52 05 95 3c 87 9e 8d 06

   hash (32 octets):  c6 c9 18 ad 2f 41 99 d5 59 8e af 01 16 cb 7a 5c
      2c 14 cb 54 78 12 18 88 8d b7 03 0d d5 0d 5e 6d

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 68 73 20 74 72
      61 66 66 69 63 20 c6 c9 18 ad 2f 41 99 d5 59 8e af 01 16 cb 7a
      5c 2c 14 cb 54 78 12 18 88 8d b7 03 0d d5 0d 5e 6d

   output (32 octets):  e2 e2 32 07 bd 93 fb 7f e4 fc 2e 29 7a fe ab
      16 0e 52 2b 5a b7 5d 64 a8 6e 75 bc ac 3f 3e 51 03

{server}  derive secret "tls13 s hs traffic":

   PRK (32 octets):  5b 4f 96 5d f0 3c 68 2c 46 e6 ee 86 c3 11 63 66
      15 a1 d2 bb b2 43 45 c2 52 05 95 3c 87 9e 8d 06

   hash (32 octets):  c6 c9 18 ad 2f 41 99 d5 59 8e af 01 16 cb 7a 5c
      2c 14 cb 54 78 12 18 88 8d b7 03 0d d5 0d 5e 6d
```



```
   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 68 73 20 74 72
      61 66 66 69 63 20 c6 c9 18 ad 2f 41 99 d5 59 8e af 01 16 cb 7a
      5c 2c 14 cb 54 78 12 18 88 8d b7 03 0d d5 0d 5e 6d

   output (32 octets):  3b 7a 83 9c 23 9e f2 bf 0b 73 05 a0 e0 c4 e5
      a8 c6 c6 93 30 a7 53 b3 08 f5 e3 a8 3a a2 ef 69 79

{server}  derive secret for master "tls13 derived":

   PRK (32 octets):  5b 4f 96 5d f0 3c 68 2c 46 e6 ee 86 c3 11 63 66
      15 a1 d2 bb b2 43 45 c2 52 05 95 3c 87 9e 8d 06

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55
```

```
      info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
         20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
         64 9b 93 4c a4 95 99 1b 78 52 b8 55

      output (32 octets):  c8 61 57 19 e2 40 37 47 b6 10 76 2c 72 b8 f4
         da 5c 60 99 57 65 d4 04 a9 d0 06 b9 b0 72 7b a5 83

   {server}  extract secret "master":

      salt (32 octets):  c8 61 57 19 e2 40 37 47 b6 10 76 2c 72 b8 f4 da
         5c 60 99 57 65 d4 04 a9 d0 06 b9 b0 72 7b a5 83

      IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      secret (32 octets):  5c 79 d1 69 42 4e 26 2b 56 32 03 62 7b e4 eb
         51 03 3f 58 8c 43 c9 ce 03 73 37 2d bc bc 01 85 a7

   {server}  send handshake record:

      payload (90 octets):  02 00 00 56 03 03 ee fc e7 f7 b3 7b a1 d1 63
         2e 96 67 78 25 dd f7 39 88 cf c7 98 25 df 56 6d c5 43 0b 9a 04
         5a 12 00 13 01 00 00 2e 00 33 00 24 00 1d 00 20 9d 3c 94 0d 89
         69 0b 84 d0 8a 60 99 3c 14 4e ca 68 4d 10 81 28 7c 83 4d 53 11
         bc f3 2b b9 da 1a 00 2b 00 02 03 04

      ciphertext (95 octets):  16 03 03 00 5a 02 00 00 56 03 03 ee fc e7
         f7 b3 7b a1 d1 63 2e 96 67 78 25 dd f7 39 88 cf c7 98 25 df 56
         6d c5 43 0b 9a 04 5a 12 00 13 01 00 00 2e 00 33 00 24 00 1d 00
         20 9d 3c 94 0d 89 69 0b 84 d0 8a 60 99 3c 14 4e ca 68 4d 10 81
         28 7c 83 4d 53 11 bc f3 2b b9 da 1a 00 2b 00 02 03 04

   {server}  derive write traffic keys for handshake data:
```



```
      PRK (32 octets):  3b 7a 83 9c 23 9e f2 bf 0b 73 05 a0 e0 c4 e5 a8
         c6 c6 93 30 a7 53 b3 08 f5 e3 a8 3a a2 ef 69 79

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  c6 6c b1 ae c5 19 df 44 c9 1e 10 99 55 11
         ac 8b
```

```
   iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

   iv output (12 octets):  f7 f6 88 4c 49 81 71 6c 2d 0d 29 a4

{server}  send a EncryptedExtensions handshake message

{server}  send a Certificate handshake message

{server}  send a CertificateVerify handshake message

{server}  calculate finished "tls13 finished":

   PRK (32 octets):  3b 7a 83 9c 23 9e f2 bf 0b 73 05 a0 e0 c4 e5 a8
      c6 c6 93 30 a7 53 b3 08 f5 e3 a8 3a a2 ef 69 79

   hash (0 octets):  (empty)

   info (18 octets):  00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65
      64 00

   output (32 octets):  a8 0c b7 d1 5d b3 4a 17 ab b0 c2 37 65 be 68
      c2 6d 3f 10 da 34 90 5b 09 99 47 e5 5e 37 db 17 b3

{server}  send a Finished handshake message

{server}  send handshake record:

   payload (657 octets):  08 00 00 24 00 22 00 0a 00 14 00 12 00 1d
      00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 1c 00 02 40
      01 00 00 00 00 0b 00 01 b9 00 00 01 b5 00 01 b0 30 82 01 ac 30
      82 01 15 a0 03 02 01 02 02 01 02 30 0d 06 09 2a 86 48 86 f7 0d
      01 01 0b 05 00 30 0e 31 0c 30 0a 06 03 55 04 03 13 03 72 73 61
      30 1e 17 0d 31 36 30 37 33 30 30 31 32 33 35 39 5a 17 0d 32 36
      30 37 33 30 30 31 32 33 35 39 5a 30 0e 31 0c 30 0a 06 03 55 04
      03 13 03 72 73 61 30 81 9f 30 0d 06 09 2a 86 48 86 f7 0d 01 01
      01 05 00 03 81 8d 00 30 81 89 02 81 81 00 b4 bb 49 8f 82 79 30
      3d 98 08 36 39 9b 36 c6 98 8c 0c 68 de 55 e1 bd b8 26 d3 90 1a
      24 61 ea fd 2d e4 9a 91 d0 15 ab bc 9a 95 13 7a ce 6c 1a f1 9e
      aa 6a f9 8c 7c ed 43 12 09 98 e1 87 a8 0e e0 cc b0 52 4b 1b 01
      8c 3e 0b 63 26 4d 44 9a 6d 38 e2 2a 5f da 43 08 46 74 80 30 53
```



```
      0e f0 46 1c 8c a9 d9 ef bf ae 8e a6 d1 d0 3e 2b d1 93 ef f0 ab
```

```
   9a 80 02 c4 74 28 a6 d3 5a 8d 88 d7 9f 7f 1e 3f 02 03 01 00 01
   a3 1a 30 18 30 09 06 03 55 1d 13 04 02 30 00 30 0b 06 03 55 1d
   0f 04 04 03 02 05 a0 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05
   00 03 81 81 00 85 aa d2 a0 e5 b9 27 6b 90 8c 65 f7 3a 72 67 17
   06 18 a5 4c 5f 8a 7b 33 7d 2d f7 a5 94 36 54 17 f2 ea e8 f8 a5
   8c 8f 81 72 f9 31 9c f3 6b 7f d6 c5 5b 80 f2 1a 03 01 51 56 72
   60 96 fd 33 5e 5e 67 f2 db f1 02 70 2e 60 8c ca e6 be c1 fc 63
   a4 2a 99 be 5c 3e b7 10 7c 3c 54 e9 b9 eb 2b d5 20 3b 1c 3b 84
   e0 a8 b2 f7 59 40 9b a3 ea c9 d9 1d 40 2d cc 0c c8 f8 96 12 29
   ac 91 87 b4 2b 4d e1 00 00 0f 00 00 84 08 04 00 80 75 40 40 d0
   dd ab 8c f0 e2 da 2b c4 99 5b 86 8a d7 45 c8 e1 56 4e 33 cd e1
   78 80 a4 23 92 cc 62 4a ee f6 b6 7b b3 f0 ae 71 d9 d5 4a 23 09
   73 1d 87 dc 59 f6 42 d7 33 be 2e b2 74 84 ad 8a 8c 8e b3 51 6a
   7a c5 7f 26 25 e2 b5 c0 88 8a 85 41 f4 e7 34 f7 3d 05 47 61 df
   1d d0 2f 0e 3e 9a 33 cf a1 0b 6e 3e b4 eb f7 ac 05 3b 01 fd ab
   bd df c5 41 33 bc d2 4c 8b bd ce b2 23 b2 aa 03 45 2a 29 14 00
   00 20 ac 86 ac bc 9c d2 5a 45 b5 7a d5 b6 4d b1 5d 44 05 cf 8c
   80 e3 14 58 3e bf 32 83 ef 9a 99 31 0c

ciphertext (679 octets):  17 03 03 02 a2 f1 0b 26 d8 fc af 67 b5
   b8 28 f7 12 12 22 16 a1 cd 14 18 74 65 b7 76 37 cb cd 78 53 91
   28 bb 93 24 6d cc a1 af 56 f1 ea a2 71 66 60 77 45 5b c5 49 65
   d8 5f 05 f9 bd 36 d6 99 61 71 eb 53 6a ff 61 3e ed dc 42 ba d5
   a2 d2 22 7c 46 06 f1 21 5f 98 0e 7a fa f5 6b d3 b8 5a 51 be 13
   00 03 10 1a 75 8d 07 7b 1c 89 1d 8e 7a 22 94 7e 5a 22 98 51 fd
   42 a9 dd 42 26 08 f8 68 27 2a bf 92 b3 d4 3f b4 6a c4 20 25 93
   46 06 7f 66 32 2f d7 08 88 56 80 f4 b4 43 3c 29 11 6f 2d fa 52
   9e 09 bb a5 3c 7c d9 20 12 17 24 80 9e ad dc c8 43 07 ef 46 fc
   51 a0 b3 3d 99 d3 9d b3 37 fc d7 61 ce 0f 2b 02 dc 73 de db 6f
   dd b7 7c 4f 80 99 bd e9 3d 5b ee 08 bc f2 13 1f 29 a2 a3 7f f0
   79 49 e8 f8 bc dd 3e 83 10 b8 bf 8b 34 44 c8 5a af 0d 2a eb 2d
   4f 36 fd 14 d5 cb 51 fc eb ff 41 8b 38 27 13 6a b9 52 9e 9a 3d
   3f 35 e4 c0 ae 74 9e a2 db c9 49 82 a1 28 1d 3e 6d aa b7 19 aa
   44 60 88 93 21 a0 08 bf 10 fa 06 ac 0c 61 cc 12 2c c9 0d 5e 22
   c0 03 0c 98 6a e8 4a 33 a0 c4 7d f1 74 bc fb d5 0b f7 8f fd f2
   40 51 ab 42 3d b6 3d 58 15 db 2f 83 00 40 f3 05 21 13 1c 98 c6
   6f 16 c3 62 ad dc e2 fb a0 60 2c f0 a7 dd df 22 e8 de f7 51 6c
   df ee 95 b4 05 6c c9 ad 38 c9 53 52 33 54 21 b5 b1 ff ba df 75
   e5 21 2f da d7 a7 5f 52 a2 80 14 86 a1 ee c3 53 95 80 be e0 e4
   b3 37 cd a6 08 5a c9 ec cd 1a 0f 1a 46 ce bf bb 5c df a3 25 1a
   c2 8c 3b c8 26 14 8c 6d 8c 1e b6 a0 6f 77 f6 ff 63 2c 6a 83 e2
   83 e8 f9 df 7c 6d ba bf 1c 6e a4 06 29 a8 5b 43 ab 0c 73 d3 4f
   9d 50 72 83 2a 10 4e da 3f 75 f5 d8 3d a6 e1 48 22 a1 8e 14 09
   9d 74 9e af d8 23 ca 2a c7 54 20 86 50 1e ca 20 6c e7 88 79 20
   00 85 73 75 7c e2 f2 30 a8 90 78 2b 99 cc 68 23 77 be ee 81 27
   56 d0 4f 90 25 13 5f b5 99 d7 46 fe fe 73 16 c9 22 ac 26 5c a0
   d2 90 21 37 5a db 63 c1 50 9c 3e 24 2d fb 92 b8 de e8 91 f7 36
```

```
      8c 40 58 39 9b 8d b9 07 5f 2d cc 82 16 19 4e 50 3b 66 52 d8 7d
      2c b4 1f 99 ad fd cc 5b e5 ec 7e 1e 63 26 ac 22 d7 0b d3 ba 65
      28 27 53 2d 66 9a ff 00 51 73 59 7f 80 39 c3 ea 49 22 d3 ec 75
      76 70 22 2f 6a c2 9b 93 e9 0d 7a d3 f6 dd 96 32 8e 42 9c fc fd
      5c ca 22 70 7f e2 d8 6a d1 dc b0 be 75 6e 8e

{server}  derive secret "tls13 c ap traffic":

   PRK (32 octets):  5c 79 d1 69 42 4e 26 2b 56 32 03 62 7b e4 eb 51
      03 3f 58 8c 43 c9 ce 03 73 37 2d bc bc 01 85 a7

   hash (32 octets):  f8 c1 9e 8c 77 c0 38 79 bb c8 eb 6d 56 e0 0d d5
      d8 6e f5 59 27 ee fc 08 e1 b0 02 b6 ec e0 5d bf

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 61 70 20 74 72
      61 66 66 69 63 20 f8 c1 9e 8c 77 c0 38 79 bb c8 eb 6d 56 e0 0d
      d5 d8 6e f5 59 27 ee fc 08 e1 b0 02 b6 ec e0 5d bf

   output (32 octets):  e2 f0 db 6a 82 e8 82 80 fc 26 f7 3c 89 85 4e
      e8 61 5e 25 df 28 b2 20 79 62 fa 78 22 26 b2 36 26

{server}  derive secret "tls13 s ap traffic":

   PRK (32 octets):  5c 79 d1 69 42 4e 26 2b 56 32 03 62 7b e4 eb 51
      03 3f 58 8c 43 c9 ce 03 73 37 2d bc bc 01 85 a7

   hash (32 octets):  f8 c1 9e 8c 77 c0 38 79 bb c8 eb 6d 56 e0 0d d5
      d8 6e f5 59 27 ee fc 08 e1 b0 02 b6 ec e0 5d bf

   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 61 70 20 74 72
      61 66 66 69 63 20 f8 c1 9e 8c 77 c0 38 79 bb c8 eb 6d 56 e0 0d
      d5 d8 6e f5 59 27 ee fc 08 e1 b0 02 b6 ec e0 5d bf

   output (32 octets):  5b 73 b1 08 d9 ac 1b 9b 0c 82 48 ca 39 26 ec
      6e 7b c4 7e 41 17 06 96 39 87 ec 11 43 5d 30 57 19

{server}  derive secret "tls13 exp master":

   PRK (32 octets):  5c 79 d1 69 42 4e 26 2b 56 32 03 62 7b e4 eb 51
      03 3f 58 8c 43 c9 ce 03 73 37 2d bc bc 01 85 a7

   hash (32 octets):  f8 c1 9e 8c 77 c0 38 79 bb c8 eb 6d 56 e0 0d d5
      d8 6e f5 59 27 ee fc 08 e1 b0 02 b6 ec e0 5d bf

   info (52 octets):  00 20 10 74 6c 73 31 33 20 65 78 70 20 6d 61 73
      74 65 72 20 f8 c1 9e 8c 77 c0 38 79 bb c8 eb 6d 56 e0 0d d5 d8
```

```
      6e f5 59 27 ee fc 08 e1 b0 02 b6 ec e0 5d bf
```



```
      output (32 octets):  b7 73 34 8a 35 a0 41 f1 19 96 89 f8 df 30 09
         7b 1d 25 7a bf 5c 0a aa 16 c8 65 10 56 b9 06 d6 c6

   {server}  derive write traffic keys for application data:

      PRK (32 octets):  5b 73 b1 08 d9 ac 1b 9b 0c 82 48 ca 39 26 ec 6e
         7b c4 7e 41 17 06 96 39 87 ec 11 43 5d 30 57 19

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  a6 88 eb b5 ac 82 6d 6f 42 d4 5c 0c c4 4b
         9b 7d

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  c1 ca d4 42 5a 43 8b 5d e7 14 83 0a

   {server}  derive read traffic keys for handshake data:

      PRK (32 octets):  e2 e2 32 07 bd 93 fb 7f e4 fc 2e 29 7a fe ab 16
         0e 52 2b 5a b7 5d 64 a8 6e 75 bc ac 3f 3e 51 03

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  26 79 a4 3e 1d 76 78 40 34 ea 17 97 d5 ad
         26 49

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  54 82 40 52 90 dd 0d 2f 81 c0 d9 42

   {client}  extract secret "early":

      salt:  (absent)

      IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
```

```
      e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

{client}  derive secret for handshake "tls13 derived":

   PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
      10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55
```



```
   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
      97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{client}  extract secret "handshake":

   salt (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
      16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

   IKM (32 octets):  81 51 d1 46 4c 1b 55 53 36 23 b9 c2 24 6a 6a 0e
      6e 7e 18 50 63 e1 4a fd af f0 b6 e1 c6 1a 86 42

   secret (32 octets):  5b 4f 96 5d f0 3c 68 2c 46 e6 ee 86 c3 11 63
      66 15 a1 d2 bb b2 43 45 c2 52 05 95 3c 87 9e 8d 06

{client}  derive secret "tls13 c hs traffic" (same as server)

{client}  derive secret "tls13 s hs traffic" (same as server)

{client}  derive secret for master "tls13 derived" (same as server)

{client}  extract secret "master" (same as server)

{client}  derive read traffic keys for handshake data:

   PRK (32 octets):  3b 7a 83 9c 23 9e f2 bf 0b 73 05 a0 e0 c4 e5 a8
      c6 c6 93 30 a7 53 b3 08 f5 e3 a8 3a a2 ef 69 79

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00
```

```
      key output (16 octets):  c6 6c b1 ae c5 19 df 44 c9 1e 10 99 55 11
         ac 8b

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  f7 f6 88 4c 49 81 71 6c 2d 0d 29 a4

   {client}  calculate finished "tls13 finished" (same as server)

   {client}  derive secret "tls13 c ap traffic" (same as server)

   {client}  derive secret "tls13 s ap traffic" (same as server)

   {client}  derive secret "tls13 exp master" (same as server)
```



```
   {client}  derive write traffic keys for handshake data (same as
      server read traffic keys)

   {client}  derive read traffic keys for application data (same as
      server write traffic keys)

   {client}  calculate finished "tls13 finished":

      PRK (32 octets):  e2 e2 32 07 bd 93 fb 7f e4 fc 2e 29 7a fe ab 16
         0e 52 2b 5a b7 5d 64 a8 6e 75 bc ac 3f 3e 51 03

      hash (0 octets):  (empty)

      info (18 octets):  00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65
         64 00

      output (32 octets):  12 1b f5 86 01 b2 ed 13 bf 14 b3 ee ac bd 9d
         a4 ba ba 1e 14 3e db 66 a1 07 79 59 60 fb d9 e2 1f

   {client}  send a Finished handshake message

   {client}  send handshake record:

      payload (36 octets):  14 00 00 20 b9 02 7a 02 04 b9 72 b5 2c de fa
         58 95 0f a1 58 0d 68 c9 cb 12 4d be 69 1a 71 78 f2 5c 55 4b 23
```

ciphertext (58 octets): 17 03 03 00 35 95 39 b4 ae 2f 87 fd 8e 61 6b 29 56 28 ea 95 3d 9e 38 58 db 27 49 70 d1 98 13 ec 13 6c ae 7d 96 e0 41 77 75 fc ab d3 d8 85 8f dc 60 24 09 12 d2 18 f5 af b2 1c

{client} derive write traffic keys for application data:

PRK (32 octets): e2 f0 db 6a 82 e8 82 80 fc 26 f7 3c 89 85 4e e8 61 5e 25 df 28 b2 20 79 62 fa 78 22 26 b2 36 26

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 88 b9 6a d6 86 c8 4b e5 5a ce 18 a5 9c ce 5c 87

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): b9 9d c5 8c d5 ff 5a b0 82 fd ad 19

{client} derive secret "tls13 res master":

PRK (32 octets): 5c 79 d1 69 42 4e 26 2b 56 32 03 62 7b e4 eb 51 03 3f 58 8c 43 c9 ce 03 73 37 2d bc bc 01 85 a7

hash (32 octets): 50 2f 86 b9 57 9e c0 53 d3 28 24 e2 78 0e f6 5c c4 37 a3 56 43 45 35 6b df 79 13 ec 3b 87 96 14

info (52 octets): 00 20 10 74 6c 73 31 33 20 72 65 73 20 6d 61 73 74 65 72 20 50 2f 86 b9 57 9e c0 53 d3 28 24 e2 78 0e f6 5c c4 37 a3 56 43 45 35 6b df 79 13 ec 3b 87 96 14

output (32 octets): f7 84 42 e1 c4 b9 d4 40 ad b6 3b e6 8f 74 a5 f3 01 94 6a 2b 2b db 36 c0 45 bb 7c f5 a9 e3 02 f5

{server} calculate finished "tls13 finished" (same as client)

{server} derive read traffic keys for application data (same as client write traffic keys)

```
{server}  derive secret "tls13 res master" (same as client)

{server}  generate resumption secret "tls13 resumption":

   PRK (32 octets):  f7 84 42 e1 c4 b9 d4 40 ad b6 3b e6 8f 74 a5 f3
      01 94 6a 2b 2b db 36 c0 45 bb 7c f5 a9 e3 02 f5

   hash (2 octets):  00 00

   info (22 octets):  00 20 10 74 6c 73 31 33 20 72 65 73 75 6d 70 74
      69 6f 6e 02 00 00

   output (32 octets):  e3 4f 01 59 72 7d 1b 8e 4c 9c 17 68 59 45 a2
      86 1f 70 dc 21 05 cb 22 4b 6d bd b3 83 28 2e f5 cf

{server}  send a NewSessionTicket handshake message

{server}  send handshake record:

   payload (205 octets):  04 00 00 c9 00 00 00 1e 2f d3 99 2f 02 00
      00 00 b2 ff 09 9f 96 76 cd ff 8b 0b f8 82 5d 00 00 00 00 79 05
      a9 d2 8e fe ef 4a 47 c6 f9 b0 6a 0c ec db 00 70 d9 20 b8 98 99
      7c 75 b7 96 36 94 3e d4 20 46 a9 61 42 bd 08 4a 04 ac fa 0c 49
      0f 45 2d 75 6d ea 02 c0 f9 27 25 9f 1f 32 31 ac 0d 54 1a 76 91
      29 b7 40 ce 38 09 08 42 b8 28 c2 7f d7 29 f5 97 37 ba 98 aa 7b
      42 e0 43 c5 da 28 f8 dc a8 59 0b 2d f4 10 d5 13 4f d6 c4 ca ca
      d8 b3 03 70 60 2a fa 35 d2 65 bf 4d 12 79 76 bb 36 db da 6a 62
      6f 02 70 e2 0e eb c7 3d 6f ca e2 b1 a0 da 12 2e e9 04 2f 76 be
      56 eb f4 1a a4 69 c3 d2 c9 da 91 97 d8 00 08 00 2a 00 04 00 00
      04 00
```



```
   ciphertext (227 octets):  17 03 03 00 de 36 80 c2 b2 10 9d 25 ca
      a2 6c 3b 06 ee a9 fd c5 cb 31 61 3b a7 02 17 65 96 da 2e 88 6b
      f6 af 93 50 7b d6 81 61 ad 9c b4 78 06 53 84 2e 10 41 ec bf 00
      88 a6 5a c4 ef 43 84 19 dd 1d 95 dd d9 bd 2a d4 48 4e 7e 16 7d
      0e 6c 00 84 48 ae 58 a0 41 87 13 b6 fc 6c 51 e4 bb 23 a5 37 fb
      75 a7 4f 73 de 31 fe 6a a0 bc 52 25 15 f8 b2 5f 89 55 42 8b 5d
      e5 ac 06 76 2c ec 22 b0 aa 78 c9 43 85 ef 8e 70 fa 24 94 5b 7c
      1f 26 85 10 87 16 89 bb bb fa f2 e7 f4 a1 92 77 02 4f 95 f1 14
      3a b1 2a 31 ec 63 ad b1 28 cb 39 07 11 fd 6d 06 a4 98 df 3e 98
      61 5d 8e b1 02 e2 33 53 b4 80 ef cc a5 e8 e0 26 7a 6d 0f e2 44
      1f 14 c8 c9 66 4a ef b2 cf ff 6a e9 e0 44 27 28 b6 a0 94 0c 1e
```

```
      82 4f da 06
```

{client} generate resumption secret "tls13 resumption" (same as server)

{client} send application_data record:

```
   payload (50 octets): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
      0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
      24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31

   ciphertext (72 octets): 17 03 03 00 43 8c 34 97 da 00 ae 02 3e 53
      c0 1b 43 24 b6 65 40 4c 1b 49 e7 8f e2 bf 4d 17 f6 34 8a e8 34
      05 51 e3 63 a0 cd 05 f2 17 9c 4f ef 5a d6 89 b5 ca e0 ba e9 4a
      dc 63 63 2e 57 1f b7 9a a9 15 44 c6 39 4d 28 a1
```

{server} send application_data record:

```
   payload (50 octets): 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
      0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
      24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31

   ciphertext (72 octets): 17 03 03 00 43 f6 5f 49 fd 2d f6 cd 23 47
      c3 d3 01 66 e3 cf dd b6 30 8a 59 06 c0 76 11 2c 6a 37 ff 1d bd
      40 6b 58 13 c0 ab d7 34 88 30 17 a6 b2 83 31 86 b1 3c 14 da 5d
      75 f3 3d 87 60 78 99 94 e2 7d 82 04 3a b8 8d 65
```

{client} send alert record:

```
   payload (2 octets): 01 00

   ciphertext (24 octets): 17 03 03 00 13 2c 21 48 16 3d 79 38 a3 5f
      6a cf 2a 66 06 f8 cb d1 d9 f2
```

{server} send alert record:

```
   payload (2 octets): 01 00
```



```
      ciphertext (24 octets): 17 03 03 00 13 f8 14 1e bd b5 ed a5 11 e0
         bc e6 39 a5 6f f9 ea 82 5a 21
```

4. Resumed 0-RTT Handshake

This handshake resumes from the handshake in Section 3. Since the server provided a session ticket that permitted 0-RTT, and the client is configured for 0-RTT, the client is able to send 0-RTT data.

{client} create an ephemeral x25519 key pair:

private key (32 octets): 53 9d 7e bf a9 6c 5c eb 7d 86 f0 b9 68 2a 1d d7 b7 b6 0d 81 c2 73 50 74 35 cd d1 b7 aa 80 05 1f

public key (32 octets): b0 31 99 c3 4d 68 2d 91 db 5f 58 96 10 f6 c0 9b ec e9 9c 23 c7 7c c6 0d 1e dd 0d 25 ed 5d be 70

{client} extract secret "early":

salt: (absent)

IKM (32 octets): e3 4f 01 59 72 7d 1b 8e 4c 9c 17 68 59 45 a2 86 1f 70 dc 21 05 cb 22 4b 6d bd b3 83 28 2e f5 cf

secret (32 octets): 04 8b 40 aa 09 ff d4 c6 76 9c 54 1a 2f 46 e2 84 66 06 f7 0d 62 a6 15 97 77 29 c5 b2 81 c7 e7 15

{client} send a ClientHello handshake message

{client} calculate finished "tls13 finished":

PRK (32 octets): 20 63 8e c4 e9 90 45 a8 bb 12 1e 86 fe 65 54 82 db b3 74 0d db f6 2d 0c bc c2 04 9c 10 c7 01 34

hash (0 octets): (empty)

info (18 octets): 00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65 64 00

output (32 octets): a8 19 28 e3 08 5c 3a 85 63 ed 82 2d a9 af 7a b7 1a c5 43 2a 5f 9d 1e 6f 71 32 f1 8b 36 e2 c7 05

{client} send handshake record:

payload (512 octets): 01 00 01 fc 03 03 88 09 d2 a3 9b f9 ae b3 83 1d 2b 32 e4 ff f9 32 15 e4 fc 4f 25 71 79 71 bd 79 e8 19 41 e3 dd 9b 00 00 06 13 01 13 03 13 02 01 00 01 cd 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 14 00 12

```
      00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 33 00
      26 00 24 00 1d 00 20 b0 31 99 c3 4d 68 2d 91 db 5f 58 96 10 f6
      c0 9b ec e9 9c 23 c7 7c c6 0d 1e dd 0d 25 ed 5d be 70 00 2a 00
      00 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06 03 02
      03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02 05 02 06 02
      02 02 00 2d 00 02 01 01 00 1c 00 02 40 01 00 15 00 57 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 29 00 dd 00 b8 00 b2 ff 09 9f 96 76 cd ff 8b 0b f8 82 5d 00
      00 00 00 79 05 a9 d2 8e fe ef 4a 47 c6 f9 b0 6a 0c ec db 00 70
      d9 20 b8 98 99 7c 75 b7 96 36 94 3e d4 20 46 a9 61 42 bd 08 4a
      04 ac fa 0c 49 0f 45 2d 75 6d ea 02 c0 f9 27 25 9f 1f 32 31 ac
      0d 54 1a 76 91 29 b7 40 ce 38 09 08 42 b8 28 c2 7f d7 29 f5 97
      37 ba 98 aa 7b 42 e0 43 c5 da 28 f8 dc a8 59 0b 2d f4 10 d5 13
      4f d6 c4 ca ca d8 b3 03 70 60 2a fa 35 d2 65 bf 4d 12 79 76 bb
      36 db da 6a 62 6f 02 70 e2 0e eb c7 3d 6f ca e2 b1 a0 da 12 2e
      e9 04 2f 76 be 56 eb f4 1a a4 69 c3 d2 c9 da 91 97 d8 2f d3 99
      32 00 21 20 3c e6 69 de de c4 4e 5e 75 53 8f cc ab 3d b0 45 fb
      5d 21 01 19 99 e1 45 12 ee 3a b3 5f 2a f4 e9
```

ciphertext (517 octets): 16 03 01 02 00 01 00 01 fc 03 03 88 09

```
      d2 a3 9b f9 ae b3 83 1d 2b 32 e4 ff f9 32 15 e4 fc 4f 25 71 79
      71 bd 79 e8 19 41 e3 dd 9b 00 00 06 13 01 13 03 13 02 01 00 01
      cd 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00
      00 0a 00 14 00 12 00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01
      03 01 04 00 33 00 26 00 24 00 1d 00 20 b0 31 99 c3 4d 68 2d 91
      db 5f 58 96 10 f6 c0 9b ec e9 9c 23 c7 7c c6 0d 1e dd 0d 25 ed
      5d be 70 00 2a 00 00 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04
      03 05 03 06 03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01
      04 02 05 02 06 02 02 02 00 2d 00 02 01 01 00 1c 00 02 40 01 00
      15 00 57 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 29 00 dd 00 b8 00 b2 ff 09 9f 96 76 cd ff
      8b 0b f8 82 5d 00 00 00 00 79 05 a9 d2 8e fe ef 4a 47 c6 f9 b0
      6a 0c ec db 00 70 d9 20 b8 98 99 7c 75 b7 96 36 94 3e d4 20 46
      a9 61 42 bd 08 4a 04 ac fa 0c 49 0f 45 2d 75 6d ea 02 c0 f9 27
      25 9f 1f 32 31 ac 0d 54 1a 76 91 29 b7 40 ce 38 09 08 42 b8 28
      c2 7f d7 29 f5 97 37 ba 98 aa 7b 42 e0 43 c5 da 28 f8 dc a8 59
      0b 2d f4 10 d5 13 4f d6 c4 ca ca d8 b3 03 70 60 2a fa 35 d2 65
      bf 4d 12 79 76 bb 36 db da 6a 62 6f 02 70 e2 0e eb c7 3d 6f ca
      e2 b1 a0 da 12 2e e9 04 2f 76 be 56 eb f4 1a a4 69 c3 d2 c9 da
      91 97 d8 2f d3 99 32 00 21 20 3c e6 69 de de c4 4e 5e 75 53 8f
      cc ab 3d b0 45 fb 5d 21 01 19 99 e1 45 12 ee 3a b3 5f 2a f4 e9
```

{client} derive secret "tls13 c e traffic":

PRK (32 octets): 04 8b 40 aa 09 ff d4 c6 76 9c 54 1a 2f 46 e2 84 66 06 f7 0d 62 a6 15 97 77 29 c5 b2 81 c7 e7 15

hash (32 octets): 34 b6 f2 ae b0 97 8e 4d f4 3a a9 0f b0 c2 8c 75 c2 f8 0a f8 e6 3a 5b 22 3b c4 a1 83 04 9b 89 b9

info (53 octets): 00 20 11 74 6c 73 31 33 20 63 20 65 20 74 72 61 66 66 69 63 20 34 b6 f2 ae b0 97 8e 4d f4 3a a9 0f b0 c2 8c 75 c2 f8 0a f8 e6 3a 5b 22 3b c4 a1 83 04 9b 89 b9

output (32 octets): cb 08 b7 85 96 5c 90 ca 74 0d 54 30 7f 9b bc 69 88 fe e7 eb 03 98 08 ed 93 da 96 36 47 d9 1c 87

{client} derive secret "tls13 e exp master":

PRK (32 octets): 04 8b 40 aa 09 ff d4 c6 76 9c 54 1a 2f 46 e2 84 66 06 f7 0d 62 a6 15 97 77 29 c5 b2 81 c7 e7 15

hash (32 octets): 34 b6 f2 ae b0 97 8e 4d f4 3a a9 0f b0 c2 8c 75 c2 f8 0a f8 e6 3a 5b 22 3b c4 a1 83 04 9b 89 b9

info (54 octets): 00 20 12 74 6c 73 31 33 20 65 20 65 78 70 20 6d 61 73 74 65 72 20 34 b6 f2 ae b0 97 8e 4d f4 3a a9 0f b0 c2 8c 75 c2 f8 0a f8 e6 3a 5b 22 3b c4 a1 83 04 9b 89 b9

output (32 octets): d9 dd b0 a3 b4 b9 0c 6a 34 7e fb d3 02 e6 6b f1 e8 f7 34 f0 e2 43 f2 b5 bb b2 a1 66 07 ac 18 b7

{client} derive write traffic keys for early application data:

PRK (32 octets): cb 08 b7 85 96 5c 90 ca 74 0d 54 30 7f 9b bc 69 88 fe e7 eb 03 98 08 ed 93 da 96 36 47 d9 1c 87

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): e8 56 97 a3 12 b9 ba e5 f9 3c 30 9b 2b ad e4 85

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 62 12 30 34 1c c0 fb fe db 55 f6 75

{client} send application_data record:

payload (6 octets): 41 42 43 44 45 46

ciphertext (28 octets): 17 03 03 00 17 7c b2 38 bd c6 0b 71 2f b1 40 ca 0f 9b 9b 8b ef c9 ff 31 31 45 75 12

{server} extract secret "early" (same as client)

{server} calculate finished "tls13 finished" (same as client)

{server} create an ephemeral x25519 key pair:

private key (32 octets): 34 68 86 bf 49 a0 43 10 79 99 c8 5a e2 71 48 e2 c1 ac a0 04 38 a6 87 df c9 bb 2c f1 17 cc cc fe

public key (32 octets): 27 e0 06 8f 6e fd 82 54 08 eb 88 c7 4e e8 8d ba 83 e3 51 ed 5a 37 49 ae 94 50 5c fb d4 e7 89 28

{server} derive secret "tls13 c e traffic" (same as client)

{server} derive secret "tls13 e exp master" (same as client)

{server} send a ServerHello handshake message

{server} derive secret for handshake "tls13 derived":

PRK (32 octets): 04 8b 40 aa 09 ff d4 c6 76 9c 54 1a 2f 46 e2 84 66 06 f7 0d 62 a6 15 97 77 29 c5 b2 81 c7 e7 15

hash (32 octets): e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

info (49 octets): 00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64 20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

```
      output (32 octets):  9e fc 79 87 0b 08 c4 c6 51 20 52 50 af 9b 83
         04 79 11 b7 83 d5 d7 67 8d 7c cc e7 18 18 9e a2 ec

   {server}  extract secret "handshake":

      salt (32 octets):  9e fc 79 87 0b 08 c4 c6 51 20 52 50 af 9b 83 04
         79 11 b7 83 d5 d7 67 8d 7c cc e7 18 18 9e a2 ec

      IKM (32 octets):  b0 66 a1 5b c1 aa ee f8 79 0e 0b 02 e6 2f 82 dc
         44 64 46 e3 7d 6d 61 22 b0 d3 b9 94 ef 11 dd 3c

      secret (32 octets):  ea d8 b8 c5 9a 15 df 29 d7 9f a4 ac 31 d5 f7
         c9 0e 2e 5c 87 d9 ea fe d1 fe 69 16 cf 2f 29 37 34

   {server}  derive secret "tls13 c hs traffic":
```



```
      PRK (32 octets):  ea d8 b8 c5 9a 15 df 29 d7 9f a4 ac 31 d5 f7 c9
         0e 2e 5c 87 d9 ea fe d1 fe 69 16 cf 2f 29 37 34

      hash (32 octets):  57 f0 ae 2e 58 8f c2 e6 e9 a1 eb d1 a6 1e 58 f9
         0c 8b 8d a1 fc 38 f0 cc 9e 9f 33 d2 21 bb ca 92

      info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 68 73 20 74 72
         61 66 66 69 63 20 57 f0 ae 2e 58 8f c2 e6 e9 a1 eb d1 a6 1e 58
         f9 0c 8b 8d a1 fc 38 f0 cc 9e 9f 33 d2 21 bb ca 92

      output (32 octets):  1f c4 90 4b fb a8 99 0c 23 53 45 e7 a7 6c fc
         78 81 a2 40 af 54 10 78 44 ce c0 51 b4 06 5b f4 c2

   {server}  derive secret "tls13 s hs traffic":

      PRK (32 octets):  ea d8 b8 c5 9a 15 df 29 d7 9f a4 ac 31 d5 f7 c9
         0e 2e 5c 87 d9 ea fe d1 fe 69 16 cf 2f 29 37 34

      hash (32 octets):  57 f0 ae 2e 58 8f c2 e6 e9 a1 eb d1 a6 1e 58 f9
         0c 8b 8d a1 fc 38 f0 cc 9e 9f 33 d2 21 bb ca 92

      info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 68 73 20 74 72
         61 66 66 69 63 20 57 f0 ae 2e 58 8f c2 e6 e9 a1 eb d1 a6 1e 58
         f9 0c 8b 8d a1 fc 38 f0 cc 9e 9f 33 d2 21 bb ca 92

      output (32 octets):  9f a7 18 12 f7 2e 9b cc b4 2b 4b 06 18 95 39
```

```
      88 3d d5 8f 98 38 78 ef 87 29 12 3b 63 ff 18 fb 06

{server}  derive secret for master "tls13 derived":

   PRK (32 octets):  ea d8 b8 c5 9a 15 df 29 d7 9f a4 ac 31 d5 f7 c9
      0e 2e 5c 87 d9 ea fe d1 fe 69 16 cf 2f 29 37 34

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  d0 83 52 8c fc 36 56 8e 69 05 c2 4b f7 3a df
      9f ac a9 90 e3 57 0d e0 35 5f f4 35 f9 53 09 b1 26

{server}  extract secret "master":

   salt (32 octets):  d0 83 52 8c fc 36 56 8e 69 05 c2 4b f7 3a df 9f
      ac a9 90 e3 57 0d e0 35 5f f4 35 f9 53 09 b1 26
```



```
   IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   secret (32 octets):  8d f1 2b 80 e8 2e f5 9b da 63 dc 17 f1 3b 4f
      a6 b8 05 5a 97 dd 2a 5a e4 57 5e c9 08 b2 7b be 29

{server}  send handshake record:

   payload (96 octets):  02 00 00 5c 03 03 22 ac 26 b0 26 b9 d5 71 70
      2d ad 44 7e 2d 5a 54 d1 5a e1 e0 6f af 78 35 8a 3e 17 7b e8 3a
      ce 94 00 13 01 00 00 34 00 29 00 02 00 00 00 33 00 24 00 1d 00
      20 27 e0 06 8f 6e fd 82 54 08 eb 88 c7 4e e8 8d ba 83 e3 51 ed
      5a 37 49 ae 94 50 5c fb d4 e7 89 28 00 2b 00 02 03 04

   ciphertext (101 octets):  16 03 03 00 60 02 00 00 5c 03 03 22 ac
      26 b0 26 b9 d5 71 70 2d ad 44 7e 2d 5a 54 d1 5a e1 e0 6f af 78
      35 8a 3e 17 7b e8 3a ce 94 00 13 01 00 00 34 00 29 00 02 00 00
      00 33 00 24 00 1d 00 20 27 e0 06 8f 6e fd 82 54 08 eb 88 c7 4e
      e8 8d ba 83 e3 51 ed 5a 37 49 ae 94 50 5c fb d4 e7 89 28 00 2b
```

```
      00 02 03 04

   {server}  derive write traffic keys for handshake data:

      PRK (32 octets):  9f a7 18 12 f7 2e 9b cc b4 2b 4b 06 18 95 39 88
         3d d5 8f 98 38 78 ef 87 29 12 3b 63 ff 18 fb 06

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  ae 83 82 f6 52 62 a0 36 0e b6 8f fb 45 15
         52 6c

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  5b 5d 18 b7 ee c7 ed 46 c3 0f c1 3a

   {server}  send a EncryptedExtensions handshake message

   {server}  calculate finished "tls13 finished":

      PRK (32 octets):  9f a7 18 12 f7 2e 9b cc b4 2b 4b 06 18 95 39 88
         3d d5 8f 98 38 78 ef 87 29 12 3b 63 ff 18 fb 06

      hash (0 octets):  (empty)

      info (18 octets):  00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65
         64 00
```



```
      output (32 octets):  4d 48 4e ab 01 74 3f 01 91 fd 0d c5 10 42 26
         64 f8 67 b6 04 68 8b 5a 2f 47 12 9c 75 a0 c1 a3 63

   {server}  send a Finished handshake message

   {server}  send handshake record:

      payload (80 octets):  08 00 00 28 00 26 00 0a 00 14 00 12 00 1d 00
         17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 1c 00 02 40 01
         00 00 00 00 00 2a 00 00 14 00 00 20 ef 49 51 b0 98 8b 89 1a 6b
         9d 71 3b f2 25 a6 7a 7b 37 c2 8e ab bd 52 30 74 bc 01 aa c3 62
         f8 e2
```

```
   ciphertext (102 octets):  17 03 03 00 61 44 c1 e3 83 6b a6 a7 ba
      0d ed 9d 4c f8 17 f3 29 79 d8 5c 8b 41 da 53 b2 09 55 80 3d 9e
      a2 e3 42 ef 1a ff d6 6a 02 87 85 e2 19 6a d6 a0 db dd 27 44 3d
      36 87 26 53 c1 96 8b 0f 9c 01 bd cf de 83 cf c1 b8 43 b7 81 90
      ab ad 0d c3 ea 30 d1 be 40 e3 ce c8 96 19 88 ce f4 95 8f d1 6b
      7f 1f 9e 47 41

{server}  derive secret "tls13 c ap traffic":

   PRK (32 octets):  8d f1 2b 80 e8 2e f5 9b da 63 dc 17 f1 3b 4f a6
      b8 05 5a 97 dd 2a 5a e4 57 5e c9 08 b2 7b be 29

   hash (32 octets):  d9 66 db 0c cf bd 43 bc 19 68 47 fe 1a 60 3f cd
      93 78 65 68 9c a8 76 03 6f 28 ea 20 60 a7 77 55

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 61 70 20 74 72
      61 66 66 69 63 20 d9 66 db 0c cf bd 43 bc 19 68 47 fe 1a 60 3f
      cd 93 78 65 68 9c a8 76 03 6f 28 ea 20 60 a7 77 55

   output (32 octets):  a8 ff a2 6f e0 c9 d1 49 3c 3d 3c 3b 32 bc a1
      80 f5 9b ba be 25 96 df f8 b2 b0 a1 46 74 0f 8b 00

{server}  derive secret "tls13 s ap traffic":

   PRK (32 octets):  8d f1 2b 80 e8 2e f5 9b da 63 dc 17 f1 3b 4f a6
      b8 05 5a 97 dd 2a 5a e4 57 5e c9 08 b2 7b be 29

   hash (32 octets):  d9 66 db 0c cf bd 43 bc 19 68 47 fe 1a 60 3f cd
      93 78 65 68 9c a8 76 03 6f 28 ea 20 60 a7 77 55

   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 61 70 20 74 72
      61 66 66 69 63 20 d9 66 db 0c cf bd 43 bc 19 68 47 fe 1a 60 3f
      cd 93 78 65 68 9c a8 76 03 6f 28 ea 20 60 a7 77 55
```



```
   output (32 octets):  51 a3 db 37 0b d9 f1 ae 7d e1 88 85 09 6b cb
      c6 1f ea 9b ce 6c cb c2 a2 76 76 4f 62 26 5a 70 9f

{server}  derive secret "tls13 exp master":
```

```
   PRK (32 octets):  8d f1 2b 80 e8 2e f5 9b da 63 dc 17 f1 3b 4f a6
      b8 05 5a 97 dd 2a 5a e4 57 5e c9 08 b2 7b be 29

   hash (32 octets):  d9 66 db 0c cf bd 43 bc 19 68 47 fe 1a 60 3f cd
      93 78 65 68 9c a8 76 03 6f 28 ea 20 60 a7 77 55

   info (52 octets):  00 20 10 74 6c 73 31 33 20 65 78 70 20 6d 61 73
      74 65 72 20 d9 66 db 0c cf bd 43 bc 19 68 47 fe 1a 60 3f cd 93
      78 65 68 9c a8 76 03 6f 28 ea 20 60 a7 77 55

   output (32 octets):  a1 13 c3 cd ff b5 f6 5d 28 21 54 d1 09 93 54
      90 a0 e3 7d bd c9 e9 ca 30 8d 36 21 e4 15 e9 7a fd

{server}  derive write traffic keys for application data:

   PRK (32 octets):  51 a3 db 37 0b d9 f1 ae 7d e1 88 85 09 6b cb c6
      1f ea 9b ce 6c cb c2 a2 76 76 4f 62 26 5a 70 9f

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

   key output (16 octets):  27 c1 35 48 44 71 94 18 ec 91 eb 0b 14 f6
      75 3a

   iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

   iv output (12 octets):  ee b3 48 83 53 db a7 3d 3a fa cd 9e

{server}  derive read traffic keys for early application data (same
   as client write traffic keys)

{client}  derive secret for handshake "tls13 derived":

   PRK (32 octets):  04 8b 40 aa 09 ff d4 c6 76 9c 54 1a 2f 46 e2 84
      66 06 f7 0d 62 a6 15 97 77 29 c5 b2 81 c7 e7 15

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55
```

output (32 octets): 9e fc 79 87 0b 08 c4 c6 51 20 52 50 af 9b 83
04 79 11 b7 83 d5 d7 67 8d 7c cc e7 18 18 9e a2 ec

{client} extract secret "handshake":

salt (32 octets): 9e fc 79 87 0b 08 c4 c6 51 20 52 50 af 9b 83 04
79 11 b7 83 d5 d7 67 8d 7c cc e7 18 18 9e a2 ec

IKM (32 octets): b0 66 a1 5b c1 aa ee f8 79 0e 0b 02 e6 2f 82 dc
44 64 46 e3 7d 6d 61 22 b0 d3 b9 94 ef 11 dd 3c

secret (32 octets): ea d8 b8 c5 9a 15 df 29 d7 9f a4 ac 31 d5 f7
c9 0e 2e 5c 87 d9 ea fe d1 fe 69 16 cf 2f 29 37 34

{client} derive secret "tls13 c hs traffic" (same as server)

{client} derive secret "tls13 s hs traffic" (same as server)

{client} derive secret for master "tls13 derived" (same as server)

{client} extract secret "master" (same as server)

{client} derive read traffic keys for handshake data:

PRK (32 octets): 9f a7 18 12 f7 2e 9b cc b4 2b 4b 06 18 95 39 88
3d d5 8f 98 38 78 ef 87 29 12 3b 63 ff 18 fb 06

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): ae 83 82 f6 52 62 a0 36 0e b6 8f fb 45 15
52 6c

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 5b 5d 18 b7 ee c7 ed 46 c3 0f c1 3a

{client} calculate finished "tls13 finished" (same as server)

{client} derive secret "tls13 c ap traffic" (same as server)

{client} derive secret "tls13 s ap traffic" (same as server)

{client} derive secret "tls13 exp master" (same as server)

{client} send a EndOfEarlyData handshake message

{client} send handshake record:

payload (4 octets): 05 00 00 00

ciphertext (26 octets): 17 03 03 00 15 77 bf ce 7f c1 91 0c fa e9 65 7a 05 f3 15 9c de f8 68 5a 30 cb

{client} derive write traffic keys for handshake data:

PRK (32 octets): 1f c4 90 4b fb a8 99 0c 23 53 45 e7 a7 6c fc 78 81 a2 40 af 54 10 78 44 ce c0 51 b4 06 5b f4 c2

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): e7 d4 94 88 a4 5c 1f 1d b4 ab 7d 7f e5 46 c9 fa

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): a2 d1 32 5b eb 51 1a 7b 4a 20 c1 0c

{client} derive read traffic keys for application data (same as server write traffic keys)

{client} calculate finished "tls13 finished":

PRK (32 octets): 1f c4 90 4b fb a8 99 0c 23 53 45 e7 a7 6c fc 78 81 a2 40 af 54 10 78 44 ce c0 51 b4 06 5b f4 c2

hash (0 octets): (empty)

info (18 octets): 00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65 64 00

output (32 octets): b5 97 08 27 aa 42 a8 db ab 2b da 4c d7 67 89 5a e6 9a a1 dc f1 b3 d9 78 a0 55 d0 79 80 74 50 11

{client} send a Finished handshake message

{client} send handshake record:

payload (36 octets): 14 00 00 20 e1 75 18 96 9c 9f 46 dc 62 94 55 ae cf e2 36 db a5 48 77 fc 3d a0 7a d5 9d 13 45 77 fd 51 6e 18

ciphertext (58 octets): 17 03 03 00 35 d0 af c0 f5 b5 5b 5c 88 3c cf 4a 46 1f 7a a1 28 47 17 89 eb 7c e4 1b b6 f0 cd 67 a9 64 16 da 6c 19 ea b0 26 b0 1d f6 89 18 58 81 46 1f 38 2f 7a 7d 63 da fa 39

{client} derive write traffic keys for application data:

PRK (32 octets): a8 ff a2 6f e0 c9 d1 49 3c 3d 3c 3b 32 bc a1 80 f5 9b ba be 25 96 df f8 b2 b0 a1 46 74 0f 8b 00

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 29 ca d2 48 96 e7 df 25 ff e0 6f cd 6c 03 69 09

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): dc 81 fc 39 54 43 9c ca e1 63 96 70

{client} derive secret "tls13 res master":

PRK (32 octets): 8d f1 2b 80 e8 2e f5 9b da 63 dc 17 f1 3b 4f a6 b8 05 5a 97 dd 2a 5a e4 57 5e c9 08 b2 7b be 29

hash (32 octets): a7 87 12 0b d8 96 6c d7 5a 05 ce 0b 9c 5b 26 da b9 6b 91 9d c3 61 a3 9e 5f d1 0a 3e 05 18 48 e4

info (52 octets): 00 20 10 74 6c 73 31 33 20 72 65 73 20 6d 61 73 74 65 72 20 a7 87 12 0b d8 96 6c d7 5a 05 ce 0b 9c 5b 26 da b9 6b 91 9d c3 61 a3 9e 5f d1 0a 3e 05 18 48 e4

output (32 octets): b0 72 82 ae e5 10 c3 e3 83 02 f4 18 a7 fa fa 9e 44 11 34 69 ae ba 27 1a a1 b6 61 ce 41 52 1c ca

{server} derive read traffic keys for handshake data:

PRK (32 octets): 1f c4 90 4b fb a8 99 0c 23 53 45 e7 a7 6c fc 78 81 a2 40 af 54 10 78 44 ce c0 51 b4 06 5b f4 c2

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): e7 d4 94 88 a4 5c 1f 1d b4 ab 7d 7f e5 46

```
      c9 fa

   iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

   iv output (12 octets):  a2 d1 32 5b eb 51 1a 7b 4a 20 c1 0c

{server}  calculate finished "tls13 finished" (same as client)

{server}  derive read traffic keys for application data (same as
   client write traffic keys)

{server}  derive secret "tls13 res master" (same as client)
```



```
{client}  send application_data record:

   payload (50 octets):  00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
      0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
      24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31

   ciphertext (72 octets):  17 03 03 00 43 c4 83 d1 89 af 82 8c ee 40
      4d cb 5a 16 64 93 50 2e d9 d0 c9 18 e7 0f d8 25 0c 5f b2 13 44
      79 6d 3a 72 bb 0a 4b 5c 59 03 c2 a7 05 6b 82 fc 17 37 7f 72 e7
      b4 6a 26 a6 97 5b 7e e3 b9 0b 2a b8 65 d4 0c 3c

{server}  send application_data record:

   payload (50 octets):  00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e
      0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23
      24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31

   ciphertext (72 octets):  17 03 03 00 43 35 da 03 f1 bd 93 ac 09 82
      d8 8e 1a 9f 6e 0e 86 81 c1 a3 4c 6e 95 ee cf ba 10 54 c5 a2 11
      00 e8 7f 2b 78 ab 1f e5 a4 3f 39 a5 8e e8 40 bf 97 f5 c9 1f 97
      3a ce 78 eb 92 f8 27 91 2f 42 31 6d a1 7b 22 b9

{client}  send alert record:

   payload (2 octets):  01 00

   ciphertext (24 octets):  17 03 03 00 13 95 2b 05 3c 66 06 d8 96 08
      89 e1 77 51 23 0e d7 8f a0 80
```

{server} send alert record:

payload (2 octets): 01 00

ciphertext (24 octets): 17 03 03 00 13 46 95 47 73 f0 bf 82 91 68 34 7b 99 0b 68 bf 73 3a f5 75

5. HelloRetryRequest

In this example, the client initiates a handshake with an X25519 [RFC7748] share. The server however prefers P-256 [FIPS186] and sends a HelloRetryRequest that requires the client to generate a key share on the P-256 curve.

{client} create an ephemeral x25519 key pair:

private key (32 octets): a8 f7 4c 62 7c 09 56 a7 89 81 aa 60 39 e1 58 56 80 f4 af 93 c6 0b 4a 9c cc 35 1f 3c 1a c9 05 c8

public key (32 octets): 28 90 65 44 eb 46 f9 bc c3 63 92 0e 28 a6 4c 72 a5 ff d1 fb f5 71 06 36 c0 5b 88 ab a0 35 38 0c

{client} send a ClientHello handshake message

{client} send handshake record:

payload (180 octets): 01 00 00 b0 03 03 8f bb 74 7c 54 ca 32 cd 2b a9 d9 26 76 15 ca 2d 28 56 8c 44 0d ce 64 e3 4a 3e f6 bc 7e 98 e9 d3 00 00 06 13 01 13 03 13 02 01 00 00 81 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 08 00 06 00 1d 00 17 00 18 00 33 00 26 00 24 00 1d 00 20 28 90 65 44 eb 46 f9 bc c3 63 92 0e 28 a6 4c 72 a5 ff d1 fb f5 71 06 36 c0 5b 88 ab a0 35 38 0c 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02 05 02 06 02 02 02 00 2d 00 02 01 01 00 1c 00 02 40 01

ciphertext (185 octets): 16 03 01 00 b4 01 00 00 b0 03 03 8f bb 74 7c 54 ca 32 cd 2b a9 d9 26 76 15 ca 2d 28 56 8c 44 0d ce 64 e3 4a 3e f6 bc 7e 98 e9 d3 00 00 06 13 01 13 03 13 02 01 00 00 81 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00

```
00 0a 00 08 00 06 00 1d 00 17 00 18 00 33 00 26 00 24 00 1d 00
20 28 90 65 44 eb 46 f9 bc c3 63 92 0e 28 a6 4c 72 a5 ff d1 fb
f5 71 06 36 c0 5b 88 ab a0 35 38 0c 00 2b 00 03 02 03 04 00 0d
00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08 05 08 06 04 01 05
01 06 01 02 01 04 02 05 02 06 02 02 02 00 2d 00 02 01 01 00 1c
00 02 40 01
```

{server} send a ServerHello handshake message

{server} send handshake record:

payload (176 octets):

```
02 00 00 ac 03 03 cf 21 ad 74 e5 9a 61 11 be 1d 8c 02 1e 65 b8
91 c2 a2 11 16 7a bb 8c 5e 07 9e 09 e2 c8 a8 33 9c 00 13 01 00
00 84 00 33 00 02 00 17 00 2c 00 74 00 72 f7 b8 f7 e4 4a 25 b1
e4 15 e3 a1 d4 00 00 00 00 65 a4 46 6b 5a a7 aa eb be d0 bc 0b
6d 96 5a 58 00 30 df ac fb a2 00 23 21 e1 2a ec 00 07 b4 da c5
d1 65 20 c4 46 f0 18 49 37 ea 29 a3 07 01 78 a7 fc 5b 0f f8 3d
b3 f6 7d 0c 13 a6 a5 df e6 b9 09 87 8b 44 ec 76 80 e7 86 75 60
fe bf ed c9 1f af 1a 87 19 1b a8 c3 c8 cd 96 2f 88 13 ff 3f 47
96 ae 00 2b 00 02 03 04
```

ciphertext (181 octets):

```
16 03 03 00 b0 02 00 00 ac 03 03 cf 21 ad 74 e5 9a 61 11 be 1d
8c 02 1e 65 b8 91 c2 a2 11 16 7a bb 8c 5e 07 9e 09 e2 c8 a8 33
9c 00 13 01 00 00 84 00 33 00 02 00 17 00 2c 00 74 00 72 f7 b8
f7 e4 4a 25 b1 e4 15 e3 a1 d4 00 00 00 00 65 a4 46 6b 5a a7 aa
eb be d0 bc 0b 6d 96 5a 58 00 30 df ac fb a2 00 23 21 e1 2a ec
00 07 b4 da c5 d1 65 20 c4 46 f0 18 49 37 ea 29 a3 07 01 78 a7
fc 5b 0f f8 3d b3 f6 7d 0c 13 a6 a5 df e6 b9 09 87 8b 44 ec 76
80 e7 86 75 60 fe bf ed c9 1f af 1a 87 19 1b a8 c3 c8 cd 96 2f
88 13 ff 3f 47 96 ae 00 2b 00 02 03 04
```


{client} create an ephemeral P-256 key pair:

private key (32 octets):

```
73 eb 34 d9 e6 f4 90 00 0d 35 bc 12 94 f1 ea 1c 3f 2b f9 95 56
0a 1f 35 a2 b9 cb 21 13 d5 48 b1
```

public key (65 octets):

```
04 35 8d 1d 9c a8 f6 79 5d fa fd 0d d3 88 14 65 67 20 14 9b bc
1b 39 8a a1 46 a2 0f 60 d6 17 db 9f 02 68 3d ac 20 ac 2c 06 a3
a5 ef a3 e2 12 49 03 d6 d2 eb a7 65 b4 42 90 1f 15 51 28 f7 e7
0e 06
```

{client} send a ClientHello handshake message

{client} send handshake record:

```
   payload (512 octets):  01 00 01 fc 03 03 8f bb 74 7c 54 ca 32 cd
      2b a9 d9 26 76 15 ca 2d 28 56 8c 44 0d ce 64 e3 4a 3e f6 bc 7e
      98 e9 d3 00 00 06 13 01 13 03 13 02 01 00 01 cd 00 00 00 0b 00
      09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 08 00 06
      00 1d 00 17 00 18 00 33 00 47 00 45 00 17 00 41 04 35 8d 1d 9c
      a8 f6 79 5d fa fd 0d d3 88 14 65 67 20 14 9b bc 1b 39 8a a1 46
      a2 0f 60 d6 17 db 9f 02 68 3d ac 20 ac 2c 06 a3 a5 ef a3 e2 12
      49 03 d6 d2 eb a7 65 b4 42 90 1f 15 51 28 f7 e7 0e 06 00 2b 00
      03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08
      05 08 06 04 01 05 01 06 01 02 01 04 02 05 02 06 02 02 02 00 2c
      00 74 00 72 f7 b8 f7 e4 4a 25 b1 e4 15 e3 a1 d4 00 00 00 00 65
      a4 46 6b 5a a7 aa eb be d0 bc 0b 6d 96 5a 58 00 30 df ac fb a2
      00 23 21 e1 2a ec 00 07 b4 da c5 d1 65 20 c4 46 f0 18 49 37 ea
      29 a3 07 01 78 a7 fc 5b 0f f8 3d b3 f6 7d 0c 13 a6 a5 df e6 b9
      09 87 8b 44 ec 76 80 e7 86 75 60 fe bf ed c9 1f af 1a 87 19 1b
      a8 c3 c8 cd 96 2f 88 13 ff 3f 47 96 ae 00 2d 00 02 01 01 00 1c
      00 02 40 01 00 15 00 af 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   ciphertext (517 octets):  16 03 03 02 00 01 00 01 fc 03 03 8f bb
      74 7c 54 ca 32 cd 2b a9 d9 26 76 15 ca 2d 28 56 8c 44 0d ce 64
      e3 4a 3e f6 bc 7e 98 e9 d3 00 00 06 13 01 13 03 13 02 01 00 01
      cd 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00
```



```
      00 0a 00 08 00 06 00 1d 00 17 00 18 00 33 00 47 00 45 00 17 00
      41 04 35 8d 1d 9c a8 f6 79 5d fa fd 0d d3 88 14 65 67 20 14 9b
      bc 1b 39 8a a1 46 a2 0f 60 d6 17 db 9f 02 68 3d ac 20 ac 2c 06
      a3 a5 ef a3 e2 12 49 03 d6 d2 eb a7 65 b4 42 90 1f 15 51 28 f7
      e7 0e 06 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06
      03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02 05 02
      06 02 02 02 00 2c 00 74 00 72 f7 b8 f7 e4 4a 25 b1 e4 15 e3 a1
```

```
         d4 00 00 00 00 65 a4 46 6b 5a a7 aa eb be d0 bc 0b 6d 96 5a 58
         00 30 df ac fb a2 00 23 21 e1 2a ec 00 07 b4 da c5 d1 65 20 c4
         46 f0 18 49 37 ea 29 a3 07 01 78 a7 fc 5b 0f f8 3d b3 f6 7d 0c
         13 a6 a5 df e6 b9 09 87 8b 44 ec 76 80 e7 86 75 60 fe bf ed c9
         1f af 1a 87 19 1b a8 c3 c8 cd 96 2f 88 13 ff 3f 47 96 ae 00 2d
         00 02 01 01 00 1c 00 02 40 01 00 15 00 af 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   {server}  extract secret "early":

      salt:  (absent)

      IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
         e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   {server}  create an ephemeral P-256 key pair:

      private key (32 octets):  22 da f5 8e bd 87 da df 82 8e 6f 8c 5d
         c0 43 df 88 be 8b 63 45 02 44 5c 5c 46 3f 4f f4 2d 37 7b

      public key (65 octets):  04 3c ff 48 7b 22 65 d1 42 f8 08 c0 65 ff
         32 b1 2c b3 a6 08 58 25 6f 15 cd de 4e 94 6a 3c b6 67 1a a9 65
         2c 31 8d 06 ec d6 5c 84 60 04 58 4a d9 79 d5 47 5c 7e 6b 9d 22
         7a 14 2c 16 da 45 ac 8b d4

   {server}  send a ServerHello handshake message

   {server}  derive secret for handshake "tls13 derived":

      PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
         10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a
```

```
   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
      97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{server}  extract secret "handshake":

   salt (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
      16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

   IKM (32 octets):  65 ab 95 4f 48 f4 18 7d bd 5f 83 6f 63 95 86 5b
      87 a4 39 98 ef ae 26 ad 24 4c ba d2 aa 2c e4 69

   secret (32 octets):  86 69 c5 a3 9b 4a fb fb 02 93 d4 a7 20 0f aa
      b7 a4 95 e9 3a 7a c3 3f 8a c5 16 24 20 04 df 28 7a

{server}  derive secret "tls13 c hs traffic":

   PRK (32 octets):  86 69 c5 a3 9b 4a fb fb 02 93 d4 a7 20 0f aa b7
      a4 95 e9 3a 7a c3 3f 8a c5 16 24 20 04 df 28 7a

   hash (32 octets):  b3 c1 a8 be 98 f4 11 09 a0 ec 84 d6 0a d0 f8 03
      cc 0e 3c d8 7a b2 9a 67 fc 17 2e 76 ee 96 69 f5

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 68 73 20 74 72
      61 66 66 69 63 20 b3 c1 a8 be 98 f4 11 09 a0 ec 84 d6 0a d0 f8
      03 cc 0e 3c d8 7a b2 9a 67 fc 17 2e 76 ee 96 69 f5

   output (32 octets):  37 7b ec 72 bf e0 e9 93 89 e5 e9 13 e2 b2 95
      9b f6 22 13 87 0f fb da 69 25 ae 17 ce de 4b 0c 01

{server}  derive secret "tls13 s hs traffic":

   PRK (32 octets):  86 69 c5 a3 9b 4a fb fb 02 93 d4 a7 20 0f aa b7
      a4 95 e9 3a 7a c3 3f 8a c5 16 24 20 04 df 28 7a

   hash (32 octets):  b3 c1 a8 be 98 f4 11 09 a0 ec 84 d6 0a d0 f8 03
      cc 0e 3c d8 7a b2 9a 67 fc 17 2e 76 ee 96 69 f5

   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 68 73 20 74 72
      61 66 66 69 63 20 b3 c1 a8 be 98 f4 11 09 a0 ec 84 d6 0a d0 f8
      03 cc 0e 3c d8 7a b2 9a 67 fc 17 2e 76 ee 96 69 f5
```

```
      output (32 octets):  19 93 fc e3 6b d1 f0 4e c1 0d 14 b6 9d 3e 12
         8e 61 35 d5 1f 62 5e 14 b7 a6 c2 15 4c 63 80 21 a7

   {server}  derive secret for master "tls13 derived":

      PRK (32 octets):  86 69 c5 a3 9b 4a fb fb 02 93 d4 a7 20 0f aa b7
         a4 95 e9 3a 7a c3 3f 8a c5 16 24 20 04 df 28 7a

      hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
         27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

      info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
         20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
         64 9b 93 4c a4 95 99 1b 78 52 b8 55

      output (32 octets):  32 25 e8 e6 82 c8 0f 84 51 c2 69 99 ca 10 99
         36 69 68 8d 8c 6f 82 82 e6 94 18 37 5b 7e 10 6d 51

   {server}  extract secret "master":

      salt (32 octets):  32 25 e8 e6 82 c8 0f 84 51 c2 69 99 ca 10 99 36
         69 68 8d 8c 6f 82 82 e6 94 18 37 5b 7e 10 6d 51

      IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      secret (32 octets):  a6 57 77 cf ab f2 b2 7d fc 68 75 6f 4e fd 2d
         f9 a3 ff 0d c3 2e c3 0e 62 5f 2e 7e 18 14 a4 d2 b9

   {server}  send handshake record:

      payload (123 octets):  02 00 00 77 03 03 3f 2c 62 94 55 ca 56 6e
         8e a2 43 7d f8 73 e2 c4 06 bc a6 1a 51 da 4d b6 cb 7e 95 63 7d
         51 42 7e 00 13 01 00 00 4f 00 33 00 45 00 17 00 41 04 3c ff 48
         7b 22 65 d1 42 f8 08 c0 65 ff 32 b1 2c b3 a6 08 58 25 6f 15 cd
         de 4e 94 6a 3c b6 67 1a a9 65 2c 31 8d 06 ec d6 5c 84 60 04 58
         4a d9 79 d5 47 5c 7e 6b 9d 22 7a 14 2c 16 da 45 ac 8b d4 00 2b
         00 02 03 04

      ciphertext (128 octets):  16 03 03 00 7b 02 00 00 77 03 03 3f 2c
         62 94 55 ca 56 6e 8e a2 43 7d f8 73 e2 c4 06 bc a6 1a 51 da 4d
         b6 cb 7e 95 63 7d 51 42 7e 00 13 01 00 00 4f 00 33 00 45 00 17
         00 41 04 3c ff 48 7b 22 65 d1 42 f8 08 c0 65 ff 32 b1 2c b3 a6
```

```
      08 58 25 6f 15 cd de 4e 94 6a 3c b6 67 1a a9 65 2c 31 8d 06 ec
      d6 5c 84 60 04 58 4a d9 79 d5 47 5c 7e 6b 9d 22 7a 14 2c 16 da
      45 ac 8b d4 00 2b 00 02 03 04
```

{server} derive write traffic keys for handshake data:

PRK (32 octets): 19 93 fc e3 6b d1 f0 4e c1 0d 14 b6 9d 3e 12 8e 61 35 d5 1f 62 5e 14 b7 a6 c2 15 4c 63 80 21 a7

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 0d d2 f3 46 9c de 17 30 9f c3 0c 61 64 8d 13 b4

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 9e 33 da a8 b6 e9 71 d3 ad 89 ce 2c

{server} send a EncryptedExtensions handshake message

{server} send a Certificate handshake message

{server} send a CertificateVerify handshake message

{server} calculate finished "tls13 finished":

PRK (32 octets): 19 93 fc e3 6b d1 f0 4e c1 0d 14 b6 9d 3e 12 8e 61 35 d5 1f 62 5e 14 b7 a6 c2 15 4c 63 80 21 a7

hash (0 octets): (empty)

info (18 octets): 00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65 64 00

output (32 octets): e2 03 13 64 a4 a5 64 fc 3f f0 da 32 3b 2b 95 c3 9b 9a be 54 8a c7 19 e8 16 3d 7c c6 9f b6 6b 4c

{server} send a Finished handshake message

{server} send handshake record:

payload (645 octets): 08 00 00 18 00 16 00 0a 00 08 00 06 00 17

```
   00 18 00 1d 00 1c 00 02 40 01 00 00 00 00 0b 00 01 b9 00 00 01
   b5 00 01 b0 30 82 01 ac 30 82 01 15 a0 03 02 01 02 02 01 02 30
   0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05 00 30 0e 31 0c 30 0a 06
   03 55 04 03 13 03 72 73 61 30 1e 17 0d 31 36 30 37 33 30 30 31
   32 33 35 39 5a 17 0d 32 36 30 37 33 30 30 31 32 33 35 39 5a 30
   0e 31 0c 30 0a 06 03 55 04 03 13 03 72 73 61 30 81 9f 30 0d 06
   09 2a 86 48 86 f7 0d 01 01 01 05 00 03 81 8d 00 30 81 89 02 81
   81 00 b4 bb 49 8f 82 79 30 3d 98 08 36 39 9b 36 c6 98 8c 0c 68
   de 55 e1 bd b8 26 d3 90 1a 24 61 ea fd 2d e4 9a 91 d0 15 ab bc
   9a 95 13 7a ce 6c 1a f1 9e aa 6a f9 8c 7c ed 43 12 09 98 e1 87
   a8 0e e0 cc b0 52 4b 1b 01 8c 3e 0b 63 26 4d 44 9a 6d 38 e2 2a
   5f da 43 08 46 74 80 30 53 0e f0 46 1c 8c a9 d9 ef bf ae 8e a6
```



```
   d1 d0 3e 2b d1 93 ef f0 ab 9a 80 02 c4 74 28 a6 d3 5a 8d 88 d7
   9f 7f 1e 3f 02 03 01 00 01 a3 1a 30 18 30 09 06 03 55 1d 13 04
   02 30 00 30 0b 06 03 55 1d 0f 04 04 03 02 05 a0 30 0d 06 09 2a
   86 48 86 f7 0d 01 01 0b 05 00 03 81 81 00 85 aa d2 a0 e5 b9 27
   6b 90 8c 65 f7 3a 72 67 17 06 18 a5 4c 5f 8a 7b 33 7d 2d f7 a5
   94 36 54 17 f2 ea e8 f8 a5 8c 8f 81 72 f9 31 9c f3 6b 7f d6 c5
   5b 80 f2 1a 03 01 51 56 72 60 96 fd 33 5e 5e 67 f2 db f1 02 70
   2e 60 8c ca e6 be c1 fc 63 a4 2a 99 be 5c 3e b7 10 7c 3c 54 e9
   b9 eb 2b d5 20 3b 1c 3b 84 e0 a8 b2 f7 59 40 9b a3 ea c9 d9 1d
   40 2d cc 0c c8 f8 96 12 29 ac 91 87 b4 2b 4d e1 00 00 0f 00 00
   84 08 04 00 80 6b b7 6f a4 24 aa d9 99 c2 72 49 23 c1 6c 5e 44
   6d 47 2e d4 2c e2 0b 66 f6 e3 3c c0 9a b6 84 09 24 30 17 45 f4
   48 f8 22 e8 cd b1 e7 1e 74 2f 41 91 8e df a3 37 54 42 11 11 6c
   33 3a 36 9f a8 97 61 07 6d d6 71 3a 28 e0 7a 22 4f c6 4d 1f dc
   8d 6f 23 01 90 05 36 f4 a9 2c 00 8d 09 9a cb 68 d8 15 9c ff f0
   ac c3 71 f8 9e 4a f0 19 b2 35 f0 c5 1d 71 a4 21 b8 ca 8d 03 36
   87 00 74 ce 7b 05 8a 14 00 00 20 d2 0d 7e 67 8b 35 c0 03 2e 96
   37 6f 7a 49 40 bc f3 20 4b 90 3e cf 90 ed af ec eb 95 f3 02 3d
   32

ciphertext (667 octets):  17 03 03 02 96 68 9c 22 eb eb c7 1d df
   1b 02 14 96 5a 39 a0 61 bf 12 af 84 c2 ee 0e 12 13 ae 3e 1c ab
   c0 ce ca c6 06 37 3e 81 eb 3f 61 55 5e e5 a4 58 bf d4 3e db e1
   f2 eb 0c b8 28 01 27 9e 02 15 8c 7b 50 3b 86 a1 42 a7 56 c4 1e
   d2 40 b8 0f e8 c4 b1 93 66 ec f1 ac 3a b7 64 f0 c5 37 7a ef 35
   6f 27 d6 01 3e af 26 ad bc 72 fc 49 4b 6e bc 9d c2 55 75 44 18
   38 cf 02 9e 73 05 72 7e f8 0d 7b 7d 51 21 2e d4 d8 8a f5 bc 1a
   80 37 8e 1c 6a 28 8e e5 14 75 7b ea b7 8a 48 af fc 89 7c 49 20
   2c fd ed 99 a7 81 05 cf 87 69 a4 c3 00 1b 81 82 66 67 03 ce c8
```

```
0b 15 a2 c4 61 68 f8 cb 44 23 70 e6 1c 4d cd f5 bc c0 25 53 f7
50 31 10 11 9f 15 0e 05 94 d5 a3 63 b2 7e 27 72 dc 96 79 24 d3
d6 ce b8 6e 7d d0 01 6b 8f 33 92 51 36 e4 69 6c 6d 43 38 4b 31
12 ec 7c 15 8a f6 88 ce 18 83 26 67 b4 ff fe 2a c4 17 4a 98 eb
fd c9 17 45 1c 96 76 a4 f3 21 f1 65 64 ec 23 90 ba 37 c3 00 b1
e7 a9 da 6c ce b2 ac 0c 45 13 5b 66 84 32 2b b2 34 f9 46 70 2a
c2 42 c7 55 7c 71 f0 ee 65 a6 c9 a7 93 24 d6 94 fe 1f 7f b2 67
ce 6e 83 22 5c 9f 10 b5 b8 8d db 25 53 5b f6 cc 73 2f c7 da 79
b8 09 28 90 82 7a 00 97 11 74 a5 f0 90 30 d0 b9 bb 5f 22 8b 08
f7 aa 2f 7c 2c 57 ac 9b 7d 69 c8 1d 56 f0 db 07 98 9e 87 4c 4e
42 0e d8 32 aa 87 4d 72 c3 c9 36 c0 85 00 f5 aa 3a 9a 9c 8f 76
f7 41 b7 dc 20 82 ab 8b 8f f4 e7 4e 8b 47 e6 6b 26 fc c6 ff bc
9a 68 b0 5b 1a db 37 bf 6e da 22 99 23 ee 4b 40 f6 3c 34 90 c6
63 f6 82 f4 12 58 25 5e 94 2a 36 7a cd 0c 7d f9 c8 7e 6a 75 5e
53 7e 7e 1a cb ba b7 b1 a4 30 b9 26 75 e4 5c 97 58 14 ed 91 7e
78 30 7a 5f 99 6b 87 47 f4 41 ca 36 93 2d 45 d5 2a 0b b1 48 6a
6f 53 75 0d 01 23 f0 8a d7 70 ca c6 8c 00 d2 84 e3 ac 09 05 80
68 ca af d4 f9 ae 46 92 04 01 cb 57 9c c4 67 ad f7 67 80 08 c5
95 32 06 51 e5 8c 92 cc 99 a6 62 9d 5f bd 57 34 ac 3f cc 34 21
```



```
5d 31 b6 09 d2 c7 86 11 00 f4 70 12 ae 8d dc 40 bd ba b9 fa 72
2a e6 cc 2a bb b8 93 14 fb 06 be 8f 2f 2b cb 65 af 5b 1e ba 49
c5 9e af 94 a1 a0 f9 33 53 f6 e2 fb 84 c9 48 0c cb 35 be 46 cb
cd 3f b4 12 64 87 f0 72 eb d8 e5 62 5d c9 aa e7 b0 7b 93 e8 de
34 21 6f
```

{server} derive secret "tls13 c ap traffic":

PRK (32 octets): a6 57 77 cf ab f2 b2 7d fc 68 75 6f 4e fd 2d f9
a3 ff 0d c3 2e c3 0e 62 5f 2e 7e 18 14 a4 d2 b9

hash (32 octets): 6c 45 a9 b1 b6 a9 d8 18 94 52 79 25 8e cc 16 fa
33 9c e6 c6 37 17 56 1c 67 ee b2 ca 27 dc d0 0e

info (54 octets): 00 20 12 74 6c 73 31 33 20 63 20 61 70 20 74 72
61 66 66 69 63 20 6c 45 a9 b1 b6 a9 d8 18 94 52 79 25 8e cc 16
fa 33 9c e6 c6 37 17 56 1c 67 ee b2 ca 27 dc d0 0e

output (32 octets): f3 72 b2 bf 29 76 71 90 a8 e0 fd 31 33 47 d8
15 14 2c 37 76 3d c1 00 78 71 91 1f 7b 5c 31 0d 40

{server} derive secret "tls13 s ap traffic":

```
      PRK (32 octets):  a6 57 77 cf ab f2 b2 7d fc 68 75 6f 4e fd 2d f9
         a3 ff 0d c3 2e c3 0e 62 5f 2e 7e 18 14 a4 d2 b9

      hash (32 octets):  6c 45 a9 b1 b6 a9 d8 18 94 52 79 25 8e cc 16 fa
         33 9c e6 c6 37 17 56 1c 67 ee b2 ca 27 dc d0 0e

      info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 61 70 20 74 72
         61 66 66 69 63 20 6c 45 a9 b1 b6 a9 d8 18 94 52 79 25 8e cc 16
         fa 33 9c e6 c6 37 17 56 1c 67 ee b2 ca 27 dc d0 0e

      output (32 octets):  a8 b8 89 78 fb a9 0f 05 7c 52 c6 77 6a 01 1a
         d5 64 bc 4d 38 ee 6c d7 45 4b a2 21 c2 89 10 08 7a

   {server}  derive secret "tls13 exp master":

      PRK (32 octets):  a6 57 77 cf ab f2 b2 7d fc 68 75 6f 4e fd 2d f9
         a3 ff 0d c3 2e c3 0e 62 5f 2e 7e 18 14 a4 d2 b9

      hash (32 octets):  6c 45 a9 b1 b6 a9 d8 18 94 52 79 25 8e cc 16 fa
         33 9c e6 c6 37 17 56 1c 67 ee b2 ca 27 dc d0 0e

      info (52 octets):  00 20 10 74 6c 73 31 33 20 65 78 70 20 6d 61 73
         74 65 72 20 6c 45 a9 b1 b6 a9 d8 18 94 52 79 25 8e cc 16 fa 33
         9c e6 c6 37 17 56 1c 67 ee b2 ca 27 dc d0 0e
```



```
      output (32 octets):  de e8 d9 7e ec e8 97 93 e4 5d 63 b4 10 18 88
         df 06 a4 d3 63 c9 d8 ff af ef 2e bd 10 64 4d bc 42

   {server}  derive write traffic keys for application data:

      PRK (32 octets):  a8 b8 89 78 fb a9 0f 05 7c 52 c6 77 6a 01 1a d5
         64 bc 4d 38 ee 6c d7 45 4b a2 21 c2 89 10 08 7a

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  df 25 5c 0d f2 0f 01 26 2c 77 1c b8 74 67
         7b 4a

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00
```

```
   iv output (12 octets):  90 89 9d 4b ab a4 31 d1 e3 1b f7 02

{server}  derive read traffic keys for handshake data:

   PRK (32 octets):  37 7b ec 72 bf e0 e9 93 89 e5 e9 13 e2 b2 95 9b
      f6 22 13 87 0f fb da 69 25 ae 17 ce de 4b 0c 01

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

   key output (16 octets):  67 b6 7b 0d c0 12 44 92 42 dd ad ff c0 b1
      7c 7e

   iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

   iv output (12 octets):  52 ac 28 15 2f f3 e1 26 02 60 08 cb

{client}  extract secret "early":

   salt:  (absent)

   IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
      e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

{client}  derive secret for handshake "tls13 derived":

   PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
      10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55
```



```
   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
      97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{client}  extract secret "handshake":
```

salt (32 octets): 6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

IKM (32 octets): 65 ab 95 4f 48 f4 18 7d bd 5f 83 6f 63 95 86 5b
87 a4 39 98 ef ae 26 ad 24 4c ba d2 aa 2c e4 69

secret (32 octets): 86 69 c5 a3 9b 4a fb fb 02 93 d4 a7 20 0f aa
b7 a4 95 e9 3a 7a c3 3f 8a c5 16 24 20 04 df 28 7a

{client} derive secret "tls13 c hs traffic" (same as server)

{client} derive secret "tls13 s hs traffic" (same as server)

{client} derive secret for master "tls13 derived" (same as server)

{client} extract secret "master" (same as server)

{client} derive read traffic keys for handshake data:

PRK (32 octets): 19 93 fc e3 6b d1 f0 4e c1 0d 14 b6 9d 3e 12 8e
61 35 d5 1f 62 5e 14 b7 a6 c2 15 4c 63 80 21 a7

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 0d d2 f3 46 9c de 17 30 9f c3 0c 61 64 8d
13 b4

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 9e 33 da a8 b6 e9 71 d3 ad 89 ce 2c

{client} calculate finished "tls13 finished" (same as server)

{client} derive secret "tls13 c ap traffic" (same as server)

{client} derive secret "tls13 s ap traffic" (same as server)

{client} derive secret "tls13 exp master" (same as server)

{client} derive write traffic keys for handshake data (same as

server read traffic keys)

{client} derive read traffic keys for application data (same as server write traffic keys)

{client} calculate finished "tls13 finished":

PRK (32 octets): 37 7b ec 72 bf e0 e9 93 89 e5 e9 13 e2 b2 95 9b f6 22 13 87 0f fb da 69 25 ae 17 ce de 4b 0c 01

hash (0 octets): (empty)

info (18 octets): 00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65 64 00

output (32 octets): 19 4b 6b 62 26 c8 11 3f e1 24 2a 2b 08 9d 39 9a 26 83 ee 49 68 d9 ff 9b de c3 dd df 25 83 a0 a6

{client} send a Finished handshake message

{client} send handshake record:

payload (36 octets): 14 00 00 20 a7 da 09 8b 9b 26 83 71 64 64 1f 9d 0d 1b de c6 e8 eb 48 35 6b e7 c0 b1 7b 6d 19 4b 4b 8f a1 fd

ciphertext (58 octets): 17 03 03 00 35 87 b5 65 69 20 5c c2 cc c4 53 67 58 88 e4 d8 79 1c 5d cf f4 26 cf 1a 88 57 84 50 54 bf 28 37 3b 9a 8e d0 99 e1 e8 31 77 fb da 25 b3 78 7a ae 3c e1 f1 a0 a7 af

{client} derive write traffic keys for application data:

PRK (32 octets): f3 72 b2 bf 29 76 71 90 a8 e0 fd 31 33 47 d8 15 14 2c 37 76 3d c1 00 78 71 91 1f 7b 5c 31 0d 40

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 06 ea a9 34 99 1d 0b 76 0d 56 9f 8e bb 79 22 8b

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 87 c0 5d f1 e8 a1 87 ba 4f e3 28 b3

{client} derive secret "tls13 res master":

PRK (32 octets): a6 57 77 cf ab f2 b2 7d fc 68 75 6f 4e fd 2d f9
a3 ff 0d c3 2e c3 0e 62 5f 2e 7e 18 14 a4 d2 b9

hash (32 octets): f6 d2 e9 99 c9 ce 6e 62 67 b3 83 3d d9 10 cd 91
92 4a f6 89 00 66 d8 51 bd 9e f2 01 65 6c d6 c8

info (52 octets): 00 20 10 74 6c 73 31 33 20 72 65 73 20 6d 61 73
74 65 72 20 f6 d2 e9 99 c9 ce 6e 62 67 b3 83 3d d9 10 cd 91 92
4a f6 89 00 66 d8 51 bd 9e f2 01 65 6c d6 c8

output (32 octets): 1f 63 61 ef 0f 9d fe 19 ac 0f eb 5d 87 51 5f
ad 41 92 67 6b 79 61 ea 85 fc 2b 31 ba a0 c1 1f fa

{server} calculate finished "tls13 finished" (same as client)

{server} derive read traffic keys for application data (same as client write traffic keys)

{server} derive secret "tls13 res master" (same as client)

{client} send alert record:

payload (2 octets): 01 00

ciphertext (24 octets): 17 03 03 00 13 a1 93 82 ba 6a cc c4 d0 df
e3 46 c6 5b b3 ff 01 95 6f 26

{server} send alert record:

payload (2 octets): 01 00

ciphertext (24 octets): 17 03 03 00 13 6a c7 95 b6 5c a3 13 33 30
22 5c c3 a8 0b 28 f2 39 d2 e9

6. Client Authentication

In this example, the server requests client authentication. The client uses a certificate with an RSA key, the server uses an ECDSA certificate with a P-256 key. Note that private keys for this example are not included in the draft.

{client} create an ephemeral x25519 key pair:

private key (32 octets): 51 51 41 c1 11 7c f2 f1 81 f0 63 41 08
da 12 41 26 df 69 36 21 2b b4 8c 0a 48 b6 86 4d 14 8a 35

```
   public key (32 octets):  8e 61 95 b8 3b ea 47 57 fc 4f c5 c9 cc 73
      2b 87 10 c0 fe 12 1f dc 3b 46 53 85 0e c0 68 bd 6a 03
```



```
{client}  send a ClientHello handshake message

{client}  send handshake record:

   payload (192 octets):  01 00 00 bc 03 03 72 be 9e 03 79 d1 64 11
      d3 5d a6 b5 56 16 bc 37 5d a6 40 55 2b ca 71 9d ae 41 90 f3 94
      39 d8 5a 00 00 06 13 01 13 03 13 02 01 00 00 8d 00 00 00 0b 00
      09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 14 00 12
      00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 33 00
      26 00 24 00 1d 00 20 8e 61 95 b8 3b ea 47 57 fc 4f c5 c9 cc 73
      2b 87 10 c0 fe 12 1f dc 3b 46 53 85 0e c0 68 bd 6a 03 00 2b 00
      03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08
      05 08 06 04 01 05 01 06 01 02 01 04 02 05 02 06 02 02 02 00 2d
      00 02 01 01 00 1c 00 02 40 01

   ciphertext (197 octets):  16 03 01 00 c0 01 00 00 bc 03 03 72 be
      9e 03 79 d1 64 11 d3 5d a6 b5 56 16 bc 37 5d a6 40 55 2b ca 71
      9d ae 41 90 f3 94 39 d8 5a 00 00 06 13 01 13 03 13 02 01 00 00
      8d 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00
      00 0a 00 14 00 12 00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01
      03 01 04 00 33 00 26 00 24 00 1d 00 20 8e 61 95 b8 3b ea 47 57
      fc 4f c5 c9 cc 73 2b 87 10 c0 fe 12 1f dc 3b 46 53 85 0e c0 68
      bd 6a 03 00 2b 00 03 02 03 04 00 0d 00 20 00 1e 04 03 05 03 06
      03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02 05 02
      06 02 02 02 00 2d 00 02 01 01 00 1c 00 02 40 01

{server}  extract secret "early":

   salt:  (absent)

   IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
      e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

{server}  create an ephemeral x25519 key pair:

   private key (32 octets):  82 0f ba 6b 13 3f a3 bb 45 4e a0 fe 61
```

```
      7e 50 3a 74 c3 09 b3 82 28 07 71 7d e1 ee 3f ee 17 27 57

   public key (32 octets):  23 dc 3e 49 2e c4 56 63 c3 ad b5 17 ec 8e
      ef a6 5b 76 c0 cf 21 21 f4 af f5 09 50 0c 05 19 7f 0a

{server}  send a ServerHello handshake message

{server}  derive secret for handshake "tls13 derived":
```



```
   PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
      10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
      97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{server}  extract secret "handshake":

   salt (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
      16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

   IKM (32 octets):  b2 0d 9a cb a0 e0 38 1d 9f f5 1e 9d 7c b8 ba 18
      a9 ba 63 7e e5 93 08 13 da 7f f8 62 e6 62 44 45

   secret (32 octets):  ba c8 e6 23 e4 82 31 e5 f0 96 4f fc 3b f3 5a
      e4 bc 65 59 1a 9e 1a cf f3 6d 18 3f d6 0a 26 bc e6

{server}  derive secret "tls13 c hs traffic":

   PRK (32 octets):  ba c8 e6 23 e4 82 31 e5 f0 96 4f fc 3b f3 5a e4
      bc 65 59 1a 9e 1a cf f3 6d 18 3f d6 0a 26 bc e6

   hash (32 octets):  58 7e dd f9 47 f8 d1 4f e6 32 6b 07 c3 11 0c b7
      33 89 d7 ba ed de 2f e3 04 7d 77 20 19 90 2e 4c
```

```
   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 68 73 20 74 72
      61 66 66 69 63 20 58 7e dd f9 47 f8 d1 4f e6 32 6b 07 c3 11 0c
      b7 33 89 d7 ba ed de 2f e3 04 7d 77 20 19 90 2e 4c

   output (32 octets):  23 03 a8 1a 55 a9 e2 92 d3 23 cd c8 9a b2 dd
      a1 63 40 f8 4f d9 dd 99 5c 72 50 c3 3e d3 82 b2 db

{server}  derive secret "tls13 s hs traffic":

   PRK (32 octets):  ba c8 e6 23 e4 82 31 e5 f0 96 4f fc 3b f3 5a e4
      bc 65 59 1a 9e 1a cf f3 6d 18 3f d6 0a 26 bc e6

   hash (32 octets):  58 7e dd f9 47 f8 d1 4f e6 32 6b 07 c3 11 0c b7
      33 89 d7 ba ed de 2f e3 04 7d 77 20 19 90 2e 4c
```



```
   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 68 73 20 74 72
      61 66 66 69 63 20 58 7e dd f9 47 f8 d1 4f e6 32 6b 07 c3 11 0c
      b7 33 89 d7 ba ed de 2f e3 04 7d 77 20 19 90 2e 4c

   output (32 octets):  e9 9c 61 c4 f3 08 86 7b f9 7f 1d 30 56 ff 11
      35 ad 33 f5 44 b5 c2 c6 79 9c a2 c7 bd d8 bb 56 d5

{server}  derive secret for master "tls13 derived":

   PRK (32 octets):  ba c8 e6 23 e4 82 31 e5 f0 96 4f fc 3b f3 5a e4
      bc 65 59 1a 9e 1a cf f3 6d 18 3f d6 0a 26 bc e6

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  cc c4 24 b2 2c e3 72 2a 86 5e 45 b8 fc 1c 98
      a6 36 9a 61 15 15 15 bb c8 4d f5 f7 3f e1 c5 e7 fe

{server}  extract secret "master":

   salt (32 octets):  cc c4 24 b2 2c e3 72 2a 86 5e 45 b8 fc 1c 98 a6
```

```
      36 9a 61 15 15 15 bb c8 4d f5 f7 3f e1 c5 e7 fe

   IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   secret (32 octets):  7a 50 b7 21 1f a2 3c 29 37 31 72 ad f8 50 39
      53 dc 76 53 af 95 0b 6b 61 9b 42 ce 1c a9 38 22 f1

{server}  send handshake record:

   payload (90 octets):  02 00 00 56 03 03 ed 3b 39 8e d9 27 26 f8 9e
      ac 52 ea 27 89 c1 00 9d d6 e2 5f 9f 3e c0 f4 00 3d a5 20 93 e4
      c9 34 00 13 01 00 00 2e 00 33 00 24 00 1d 00 20 23 dc 3e 49 2e
      c4 56 63 c3 ad b5 17 ec 8e ef a6 5b 76 c0 cf 21 21 f4 af f5 09
      50 0c 05 19 7f 0a 00 2b 00 02 03 04

   ciphertext (95 octets):  16 03 03 00 5a 02 00 00 56 03 03 ed 3b 39
      8e d9 27 26 f8 9e ac 52 ea 27 89 c1 00 9d d6 e2 5f 9f 3e c0 f4
      00 3d a5 20 93 e4 c9 34 00 13 01 00 00 2e 00 33 00 24 00 1d 00
      20 23 dc 3e 49 2e c4 56 63 c3 ad b5 17 ec 8e ef a6 5b 76 c0 cf
      21 21 f4 af f5 09 50 0c 05 19 7f 0a 00 2b 00 02 03 04

{server}  derive write traffic keys for handshake data:
```



```
   PRK (32 octets):  e9 9c 61 c4 f3 08 86 7b f9 7f 1d 30 56 ff 11 35
      ad 33 f5 44 b5 c2 c6 79 9c a2 c7 bd d8 bb 56 d5

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

   key output (16 octets):  61 a2 08 f9 c7 7f 35 96 9e 7f 1e 0e a2 75
      4c 92

   iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

   iv output (12 octets):  08 a7 d2 9a d2 4b bf 51 1e a2 dd 45

{server}  send a EncryptedExtensions handshake message

{server}  send a CertificateRequest handshake message

{server}  send a Certificate handshake message
```

```
{server}  send a CertificateVerify handshake message

{server}  calculate finished "tls13 finished":

   PRK (32 octets):  e9 9c 61 c4 f3 08 86 7b f9 7f 1d 30 56 ff 11 35
      ad 33 f5 44 b5 c2 c6 79 9c a2 c7 bd d8 bb 56 d5

   hash (0 octets):  (empty)

   info (18 octets):  00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65
      64 00

   output (32 octets):  9f 46 ac 32 80 c8 66 da b9 27 45 b6 af ec 7c
      b3 5a 58 1a 4a 6c 8e 5e 09 a4 9c 96 d0 ad 30 2e 34

{server}  send a Finished handshake message

{server}  send handshake record:

   payload (516 octets):  08 00 00 24 00 22 00 0a 00 14 00 12 00 1d
      00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 1c 00 02 40
      01 00 00 00 00 0d 00 00 27 00 00 24 00 0d 00 20 00 1e 04 03 05
      03 06 03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02 01 04 02
      05 02 06 02 02 02 0b 00 01 3b 00 00 01 37 00 01 32 30 82 01 2e
      30 81 d5 a0 03 02 01 02 02 01 07 30 0a 06 08 2a 86 48 ce 3d 04
      03 02 30 13 31 11 30 0f 06 03 55 04 03 13 08 65 63 64 73 61 32
      35 36 30 1e 17 0d 31 36 30 37 33 30 30 31 32 34 30 30 5a 17 0d
      32 36 30 37 33 30 30 31 32 34 30 30 5a 30 13 31 11 30 0f 06 03
      55 04 03 13 08 65 63 64 73 61 32 35 36 30 59 30 13 06 07 2a 86
      48 ce 3d 02 01 06 08 2a 86 48 ce 3d 03 01 07 03 42 00 04 08 d5
```



```
      30 16 15 75 f4 cf e7 f1 54 ee 34 48 18 00 86 00 1e 88 43 1a 79
      ee 62 ee 6e 2f 83 ef 38 ba 61 e9 fb 37 f3 4e 00 7a 7d f4 d2 f5
      b5 6d 1f 04 ec e4 5d 62 1f 46 84 06 f5 c3 a1 51 58 94 8d d0 a3
      1a 30 18 30 09 06 03 55 1d 13 04 02 30 00 30 0b 06 03 55 1d 0f
      04 04 03 02 07 80 30 0a 06 08 2a 86 48 ce 3d 04 03 02 03 48 00
      30 45 02 21 00 df 30 fd 45 07 f5 ed d2 2c 1a 6f f8 6d b4 79 ca
      69 3f ee ca 3b 71 b3 f9 ef 55 6b 29 37 c0 59 4d 02 20 62 e2 a4
      72 50 d3 20 fe a8 3c 7e 2d cb 5b 76 a5 0e 02 00 c0 9a db d1 3f
      ee 94 6e 51 3e 01 1d 11 00 00 0f 00 00 4a 04 03 00 46 30 44 02
      20 4e c5 5a 94 22 b9 26 82 ac f6 01 da 8e ad dc a8 43 17 0c 52
      94 cb b0 92 64 60 09 a2 22 8f c6 3d 02 20 33 61 b0 78 aa 93 db
```

```
      6e 9c 22 ad f1 88 5b 9e 0a 3e d4 ec dd 5c ef dc ce 63 f9 99 84
      82 0b 23 ee 14 00 00 20 65 0d bb 4b 5a 6a ce 4e 23 5c 3a 3a 39
      06 09 41 fc 25 37 58 6e 9b 56 27 2e 5f d1 31 ca 1f d2 74

   ciphertext (538 octets):  17 03 03 02 15 3a cf 25 29 62 4c 10 c3
      30 42 26 01 83 6e f0 93 ef ff c9 21 c2 60 9e 77 58 42 c4 65 ea
      a3 2c ca 23 34 06 4c 8d d8 53 96 ba 07 a8 6b d0 83 28 bc 07 e1
      f8 96 9d 93 09 68 79 a8 ee d4 af 92 e3 e3 ea 74 63 28 d6 40 22
      04 a5 9c a9 9c a8 2d 42 18 f0 85 10 60 ab ca 1e d6 c9 24 d6 49
      a1 6f 4c 5f 59 37 a6 de dd 36 de aa b7 25 ff 5c ab 8d 05 10 cc
      4d a2 c4 b7 57 7a 06 2a f1 5a 89 f7 ca 9f 8e ae 62 cf ea 55 6c
      c0 51 be ed c6 db ac 7f b2 1d a9 10 e7 07 5b 39 7c 32 f7 a5 a5
      0c e7 e8 22 9a 7c f5 db 31 8e f9 be 2a af 45 04 0d 15 96 aa 72
      d7 99 81 3b 79 37 db 78 dc cc df 5c 1a b0 bb ad 95 29 34 f2 a8
      e3 0f e2 60 2b 72 d0 11 8e fb 24 02 0c 0f 35 b1 4c bd af 1a b6
      9e 3e 6b a9 f5 1c db 02 9a 88 11 0d 97 59 26 af f0 ba 32 b2 15
      1b a6 52 db 21 ed eb a4 6e ba 90 f0 d5 51 8c e1 1c 9e 48 61 34
      ee 18 6e 98 f2 0c 06 67 93 19 5a 16 7a 38 f9 ae 57 2d 66 4b 84
      46 09 36 ca f7 fd 83 58 33 0a 99 a0 41 b5 d6 3d db 52 2a e4 20
      bd 46 e0 7a b1 da 63 4f 43 d3 c2 d6 46 cf df 0d 07 cc e4 1e ed
      c7 98 a0 ad 3d 98 51 52 40 48 0c 02 13 b1 87 37 2d 8d a1 d3 aa
      42 9f f8 20 94 34 b0 a5 a1 44 8c d6 30 1e c6 37 5e 5f f6 d9 26
      55 d1 ae 13 49 97 ef 3b 97 34 f3 89 6e 5d 2f b4 ce 0c 90 d8 d9
      ea b9 67 da f2 0f 95 05 71 2e e3 6a 33 48 6f 05 72 2a 0b 9f a7
      d8 f6 77 bd 9b 2a b2 45 97 ff 68 0b 2d 51 e7 20 f1 99 6a 58 fa
      7f 46 0a 1d 60 6d fb 7a e6 b1 22 e7 a0 9d a4 cc 92 55 dc 82 99
      15 b4 be db f1 66 2d 0f f4 56 22 a4 cf 75 0e 41 cd c6 32 a1 e0
      4c 07 2f e9 2d 32 9a 26 3f 67 62 be ad 32 31 65 92 b5 01 2d 28
      07 a2 12 17 ae 83 34 59 00 f1 f4 cb 1c 7a 77 05 27 20 60 fb 35
      12 86 16 8a ce d6 be 48 23 6a 6b c6 e6 88 f6 9d 3a 09 3d d4 89

{server}  derive secret "tls13 c ap traffic":

   PRK (32 octets):  7a 50 b7 21 1f a2 3c 29 37 31 72 ad f8 50 39 53
      dc 76 53 af 95 0b 6b 61 9b 42 ce 1c a9 38 22 f1
```



```
   hash (32 octets):  95 7f 54 ae 99 e3 22 ae 51 0d 51 4d 30 73 1b 0e
      7f f1 71 0f 69 0a 0b 0c 28 6a 66 0e c4 86 69 d7

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 61 70 20 74 72
```

```
      61 66 66 69 63 20 95 7f 54 ae 99 e3 22 ae 51 0d 51 4d 30 73 1b
      0e 7f f1 71 0f 69 0a 0b 0c 28 6a 66 0e c4 86 69 d7

   output (32 octets):  e6 47 85 57 d7 f3 3b b2 77 01 be 74 7f 2f bf
      00 72 e4 91 4f 96 7a 8a b7 20 c9 36 7f f6 61 49 2a

{server}  derive secret "tls13 s ap traffic":

   PRK (32 octets):  7a 50 b7 21 1f a2 3c 29 37 31 72 ad f8 50 39 53
      dc 76 53 af 95 0b 6b 61 9b 42 ce 1c a9 38 22 f1

   hash (32 octets):  95 7f 54 ae 99 e3 22 ae 51 0d 51 4d 30 73 1b 0e
      7f f1 71 0f 69 0a 0b 0c 28 6a 66 0e c4 86 69 d7

   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 61 70 20 74 72
      61 66 66 69 63 20 95 7f 54 ae 99 e3 22 ae 51 0d 51 4d 30 73 1b
      0e 7f f1 71 0f 69 0a 0b 0c 28 6a 66 0e c4 86 69 d7

   output (32 octets):  2e 5d c3 82 75 26 7f 49 ae bd 06 3b 4c 22 70
      5d 41 7f 79 b0 4e 63 7c 93 d3 e3 2a 7d 54 6e 2e b3

{server}  derive secret "tls13 exp master":

   PRK (32 octets):  7a 50 b7 21 1f a2 3c 29 37 31 72 ad f8 50 39 53
      dc 76 53 af 95 0b 6b 61 9b 42 ce 1c a9 38 22 f1

   hash (32 octets):  95 7f 54 ae 99 e3 22 ae 51 0d 51 4d 30 73 1b 0e
      7f f1 71 0f 69 0a 0b 0c 28 6a 66 0e c4 86 69 d7

   info (52 octets):  00 20 10 74 6c 73 31 33 20 65 78 70 20 6d 61 73
      74 65 72 20 95 7f 54 ae 99 e3 22 ae 51 0d 51 4d 30 73 1b 0e 7f
      f1 71 0f 69 0a 0b 0c 28 6a 66 0e c4 86 69 d7

   output (32 octets):  c5 10 a7 cd 37 4a 95 c4 47 ba 18 53 71 7b a6
      02 25 11 6c 89 2f 2b 62 86 26 28 a5 72 df 54 68 92

{server}  derive write traffic keys for application data:

   PRK (32 octets):  2e 5d c3 82 75 26 7f 49 ae bd 06 3b 4c 22 70 5d
      41 7f 79 b0 4e 63 7c 93 d3 e3 2a 7d 54 6e 2e b3

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00
```

key output (16 octets): 3f d4 15 17 e6 ab 77 a2 e8 2d 51 f0 34 fc 8c 21

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 8f 51 67 7a 4e 55 3e ce e0 2c c3 48

{server} derive read traffic keys for handshake data:

PRK (32 octets): 23 03 a8 1a 55 a9 e2 92 d3 23 cd c8 9a b2 dd a1 63 40 f8 4f d9 dd 99 5c 72 50 c3 3e d3 82 b2 db

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 2f b3 45 4b aa 32 08 04 f1 46 3b 6d 86 9e 5c 6e

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 8d 74 fa ab ae 3d cf 20 6d 04 dc f8

{client} extract secret "early":

salt: (absent)

IKM (32 octets): 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

secret (32 octets): 33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

{client} derive secret for handshake "tls13 derived":

PRK (32 octets): 33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

hash (32 octets): e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

info (49 octets): 00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64 20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

output (32 octets): 6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{client} extract secret "handshake":

salt (32 octets): 6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

IKM (32 octets): b2 0d 9a cb a0 e0 38 1d 9f f5 1e 9d 7c b8 ba 18 a9 ba 63 7e e5 93 08 13 da 7f f8 62 e6 62 44 45

secret (32 octets): ba c8 e6 23 e4 82 31 e5 f0 96 4f fc 3b f3 5a e4 bc 65 59 1a 9e 1a cf f3 6d 18 3f d6 0a 26 bc e6

{client} derive secret "tls13 c hs traffic" (same as server)

{client} derive secret "tls13 s hs traffic" (same as server)

{client} derive secret for master "tls13 derived" (same as server)

{client} extract secret "master" (same as server)

{client} derive read traffic keys for handshake data:

PRK (32 octets): e9 9c 61 c4 f3 08 86 7b f9 7f 1d 30 56 ff 11 35 ad 33 f5 44 b5 c2 c6 79 9c a2 c7 bd d8 bb 56 d5

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): 61 a2 08 f9 c7 7f 35 96 9e 7f 1e 0e a2 75 4c 92

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): 08 a7 d2 9a d2 4b bf 51 1e a2 dd 45

{client} calculate finished "tls13 finished" (same as server)

{client} derive secret "tls13 c ap traffic" (same as server)

{client} derive secret "tls13 s ap traffic" (same as server)

{client} derive secret "tls13 exp master" (same as server)

{client} derive write traffic keys for handshake data (same as

server read traffic keys)

{client} derive read traffic keys for application data (same as server write traffic keys)

{client} send a Certificate handshake message

{client} send a CertificateVerify handshake message

{client} calculate finished "tls13 finished":

PRK (32 octets): 23 03 a8 1a 55 a9 e2 92 d3 23 cd c8 9a b2 dd a1 63 40 f8 4f d9 dd 99 5c 72 50 c3 3e d3 82 b2 db

hash (0 octets): (empty)

info (18 octets): 00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65 64 00

output (32 octets): 9e 53 42 bd 39 7f ac 99 c3 40 bd 4a 58 0f 63 20 49 8a 4f 63 6a 61 da 92 7a a2 ef 20 75 e9 74 86

{client} send a Finished handshake message

{client} send handshake record:

payload (623 octets): 0b 00 01 bf 00 00 01 bb 00 01 b6 30 82 01 b2 30 82 01 1b a0 03 02 01 02 02 01 01 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05 00 30 11 31 0f 30 0d 06 03 55 04 03 13 06 63 6c 69 65 6e 74 30 1e 17 0d 31 36 30 37 33 30 30 31 32 33 35 39 5a 17 0d 32 36 30 37 33 30 30 31 32 33 35 39 5a 30 11 31 0f 30 0d 06 03 55 04 03 13 06 63 6c 69 65 6e 74 30 81 9f 30 0d 06 09 2a 86 48 86 f7 0d 01 01 01 05 00 03 81 8d 00 30 81 89 02 81 81 00 c3 81 75 e0 04 a6 8d 09 3f 82 3b 9c 37 9d 20 1f bc 0b b7 a1 c7 91 90 5e 3f bf 76 84 7e 44 e7 51 eb bc d3 60 bd 94 5c 81 e5 22 2b cc 88 46 d3 a8 a0 f9 3e 9b f5 be ba bd 92 ed f1 de 1f f1 90 21 70 3e 7a b6 c0 90 15 13 f9 7e 39 b1 11 f0 9c 93 48 97 1c 7b 21 19 84 a7 54 cd 45 fe 09 5a f0 ea 42 36 82 9b cc f7 a7 fe 9b 28 88 e7 8a b4 77 69 0a 5b 9e 1c cb e9 1c 6a 4a 0f 97 a7 e0 28 42 01 02 03 01 00 01 a3 1a 30 18 30 09 06 03 55 1d 13 04 02 30 00 30 0b 06 03 55 1d 0f 04 04 03 02 07 80 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05 00 03 81 81 00 1a 7a 5a 01 85 32 b0 22

```
   af 07 67 d4 86 16 0c ff 2d 16 7a 19 15 d2 38 35 b5 45 94 91 6d
   c6 80 be 5d 2e 62 60 76 c5 d5 27 22 eb cc 77 5d 7d 99 f9 80 be
   2f c9 4d 34 ac f6 cc 00 ba 90 cb cf b0 60 8a a1 e7 e3 97 1e f0
   c0 7a 41 d4 7a d8 34 5d 1f 81 fe 41 8a 1c f4 10 54 42 9f d2 17
   bd 77 7d c1 cf 08 f0 5d f9 07 99 c6 59 36 1e 0f 1a 8e e4 ac 0f
   78 97 42 0b db c8 23 da 80 a2 f2 ba 23 08 1c 00 00 0f 00 00 84
   08 04 00 80 bc cd 87 0a 6d 51 75 ab 6a 97 3f 99 0f 44 33 b9 f4
   ed ea 6a a9 4c e5 c4 a9 0a 07 0f eb b8 9e 1c f5 24 62 d6 a0 5e
   62 1b 81 96 24 eb 9b f7 57 3a 08 bb 75 3d 4a 19 43 34 59 62 19
   68 75 04 54 05 6f 3d 7c e1 22 7f c2 9e 12 31 36 3e 4e ed 5f e0
   f4 93 83 7e f6 fe 4a 63 19 52 0b 63 9a ff e7 75 ae 41 76 bb bf
   69 13 b3 a1 a6 77 a0 35 6f 3c 0f 95 3d 35 77 fb 53 76 13 eb af
   84 8e 6a ee b2 1e 14 00 00 20 97 96 f8 14 93 a1 49 f5 37 f9 9b
   3c 4c f8 55 a0 88 5c 64 10 ff a1 db 0e 25 f3 43 a5 ff b5 1d 60
```



```
ciphertext (645 octets):  17 03 03 02 80 38 0f e4 54 42 85 14 4f
   66 58 7c 3f ee 90 97 e2 e5 f4 cf ad 97 31 dc 59 62 36 7e 0f 73
   ea a8 c3 16 51 cf fc da 0c 7f 2a 85 d7 46 36 85 7e 61 91 9e 7a
   3e 1a dd 24 b1 d0 8f 37 35 04 36 f5 d2 96 78 43 6f 6a df 4e 4e
   46 f9 fb 0c 79 da 40 cb 43 dd 82 50 a5 fa bc 61 cd b3 9a 4c 3d
   31 59 6c e3 1b 4c a9 4c 77 16 f6 f8 0d 09 26 80 d6 ce bf e5 c5
   cc 0e 51 15 ff 10 a6 80 1d 82 07 f4 ec ea a8 82 02 e1 bd 55 ab
   b0 ec aa 4f 0e 41 af 70 54 e0 ff df 76 4a 84 cd 01 be a2 0f d7
   b4 91 e5 c1 20 d9 93 31 4c bd 43 55 65 25 3f b2 4b 6e 67 85 ea
   79 8f 86 2c fe 0d 01 de 13 d5 f0 d8 f3 f8 d2 75 5c 1b 4d 46 d1
   d6 a3 b2 43 ea 8b 45 12 51 2e aa 64 27 3a 84 36 3c cc 93 69 a5
   3a 0b 60 09 d4 47 23 a8 f5 aa 9d 8b c9 37 1f b0 da dc 45 16 fc
   9f 84 2d 2e 3d 89 15 39 3d 2b fa db 11 82 0f 74 2d 94 6a 2a fa
   01 4f df d7 da 08 1b 86 26 7c 3c 62 95 7e 91 83 13 3e d8 7f fe
   9e 88 3e 7b 69 8e f9 09 30 ad 93 b4 e6 b3 72 bd ca 6d 77 e1 ed
   20 71 40 2b eb d8 3f 4b 74 94 a8 02 df f2 ab d1 84 d8 c3 9e 6f
   c6 4a 94 85 a3 18 f6 8b cb a3 7d 9a f9 8b 61 e7 b5 4b 2a 48 71
   9d 41 41 9e 5b b7 03 98 49 3a e4 a4 7f 45 f1 61 22 53 15 4d da
   bd b8 c6 a3 f7 1d d6 93 69 bd fe a1 af 5c b6 35 d1 8b 97 38 24
   8b cb 9c fc 61 08 e0 90 2b 86 f6 26 03 19 43 15 ae 51 d3 ac b1
   2d 06 b7 d9 86 14 bf 8e 93 f2 d4 d4 a5 6f e8 2d 09 12 e1 57 bc
   c5 28 7b 5f 1e f9 a9 db d8 a0 80 19 5f 6b 15 5a f9 16 7c ca 41
   45 35 4c 03 19 51 ab e3 73 4a 49 84 01 37 70 64 a0 d0 08 76 4d
   75 9f d1 c8 ea 7b d3 6b ec a2 23 e4 86 fc e9 89 9c de fe a6 95
   ba 7d da f2 3f 80 6b 09 ff ef 81 47 87 c7 71 ba 60 90 08 13 4d
   d4 51 1e 26 5f 78 b6 25 91 74 76 42 7b ed b7 9a 50 c3 b7 58 01
```

```
      07 5d 13 3f 2e 07 15 e7 1f c0 07 89 eb dc ce f6 b8 cd f2 5d a4
      19 bc 00 28 74 4a 75 ba ab 09 25 4a 2b b2 19 81 d1 15 64 64 22
      98 4f 79 eb c7 0a f1 39 0a b1 a2 ac 38 5c 6a d1 28 fd 9d e3 bf
      7d be e6 0f 7f e6 d0 09 e7 ce d6 8b d3 7a 06 fd db 83 5f 8e 56
      fc eb 59 4f 74 8a 1d a1 e7 7b ea 51 fb 3e 40 b7 b4 70 12 89 ef
      7b 37

{client}  derive write traffic keys for application data:

   PRK (32 octets):  e6 47 85 57 d7 f3 3b b2 77 01 be 74 7f 2f bf 00
      72 e4 91 4f 96 7a 8a b7 20 c9 36 7f f6 61 49 2a

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

   key output (16 octets):  db 34 ce df e4 fc db 0e d7 00 41 8f dd 96
      b2 c7

   iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

   iv output (12 octets):  e1 7e 53 1a ba 3c fa 7f 0f ec 8b f5

{client}  derive secret "tls13 res master":
```



```
   PRK (32 octets):  7a 50 b7 21 1f a2 3c 29 37 31 72 ad f8 50 39 53
      dc 76 53 af 95 0b 6b 61 9b 42 ce 1c a9 38 22 f1

   hash (32 octets):  41 bf 98 c7 24 79 cf cf 1d 49 9d c2 d6 a8 44 c1
      7f 49 1e b9 a0 78 21 08 78 b5 5a e5 26 29 94 60

   info (52 octets):  00 20 10 74 6c 73 31 33 20 72 65 73 20 6d 61 73
      74 65 72 20 41 bf 98 c7 24 79 cf cf 1d 49 9d c2 d6 a8 44 c1 7f
      49 1e b9 a0 78 21 08 78 b5 5a e5 26 29 94 60

   output (32 octets):  c4 11 50 3f ea fa f0 d7 0a 77 c6 81 3d b0 42
      4e f5 f4 ce f4 b5 e2 4d b7 65 f8 79 d3 7f c5 b6 af

{server}  calculate finished "tls13 finished" (same as client)

{server}  derive read traffic keys for application data (same as
   client write traffic keys)

{server}  derive secret "tls13 res master" (same as client)
```

{client} send alert record:

payload (2 octets): 01 00

ciphertext (24 octets): 17 03 03 00 13 d1 3c 7f 7d 16 11 b4 09 df 45 77 ca 2b e5 a8 a2 8f 33 30

{server} send alert record:

payload (2 octets): 01 00

ciphertext (24 octets): 17 03 03 00 13 37 bb 98 68 73 81 3c 79 25 aa 29 51 e1 21 b0 58 57 f7 8f

7. Compatibility Mode

This example shows use of the handshake with the client requesting that the server use compatibility mode as defined in Appendix D.4 of [TLS13].

{client} create an ephemeral x25519 key pair:

private key (32 octets): ea e2 7f 11 4d a0 68 f8 b3 47 2e 62 88 00 e8 b9 c2 58 13 58 13 6e bb e7 74 38 cb 4f 4b e2 d1 b4

public key (32 octets): d5 15 42 62 5f 25 a9 2d 44 a3 aa de f5 9c a8 49 ad 2f 8e fa 9f 04 b8 f5 da b4 02 ac bc 57 1f 16

{client} send a ClientHello handshake message

{client} send handshake record:

payload (224 octets): 01 00 00 dc 03 03 37 b0 76 d2 fa 50 94 39 5e 99 71 d7 53 c3 c4 cf 07 56 b9 40 70 13 cb ca c7 f4 4a c3 28 13 f6 0f 20 91 41 b7 89 83 d3 67 a0 fe 97 08 df 32 f5 b9 88 8f e5 9e de 4e 61 2c f6 bd b1 fb be e6 f9 ef fe 00 06 13 01 13 03 13 02 01 00 00 8d 00 00 00 0b 00 09 00 00 06 73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 14 00 12 00 1d 00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 33 00 26 00 24 00 1d 00 20 d5 15 42 62 5f 25 a9 2d 44 a3 aa de f5 9c a8 49 ad 2f 8e fa 9f 04 b8 f5

```
      da b4 02 ac bc 57 1f 16 00 2b 00 03 02 03 04 00 0d 00 20 00 1e
      04 03 05 03 06 03 02 03 08 04 08 05 08 06 04 01 05 01 06 01 02
      01 04 02 05 02 06 02 02 02 00 2d 00 02 01 01 00 1c 00 02 40 01

   ciphertext (229 octets):  16 03 01 00 e0 01 00 00 dc 03 03 37 b0
      76 d2 fa 50 94 39 5e 99 71 d7 53 c3 c4 cf 07 56 b9 40 70 13 cb
      ca c7 f4 4a c3 28 13 f6 0f 20 91 41 b7 89 83 d3 67 a0 fe 97 08
      df 32 f5 b9 88 8f e5 9e de 4e 61 2c f6 bd b1 fb be e6 f9 ef fe
      00 06 13 01 13 03 13 02 01 00 00 8d 00 00 00 0b 00 09 00 00 06
      73 65 72 76 65 72 ff 01 00 01 00 00 0a 00 14 00 12 00 1d 00 17
      00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 33 00 26 00 24 00
      1d 00 20 d5 15 42 62 5f 25 a9 2d 44 a3 aa de f5 9c a8 49 ad 2f
      8e fa 9f 04 b8 f5 da b4 02 ac bc 57 1f 16 00 2b 00 03 02 03 04
      00 0d 00 20 00 1e 04 03 05 03 06 03 02 03 08 04 08 05 08 06 04
      01 05 01 06 01 02 01 04 02 05 02 06 02 02 02 00 2d 00 02 01 01
      00 1c 00 02 40 01

{server}  extract secret "early":

   salt:  (absent)

   IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

   secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
      e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

{server}  create an ephemeral x25519 key pair:

   private key (32 octets):  6f fc 0f 52 08 bb f6 73 4b 5f 95 23 7d
      3d 48 0a 08 fc e9 89 e6 1c 2f 4d 71 6b 5b e4 4d 66 90 7e

   public key (32 octets):  ab 16 0e 03 51 0f a0 3f d5 bd 6e 7a 94 f4
      00 31 16 35 cd 69 87 2e a6 e4 8a 08 71 5e e3 f0 24 2e

{server}  send a ServerHello handshake message
```



```
{server}  send handshake record:

   payload (122 octets):  02 00 00 76 03 03 32 a4 2f 56 c8 b8 59 cc
      5d 80 f2 7f 48 d0 f2 96 d3 a5 bb 8e 05 28 08 11 14 de 8c e3 84
      d7 e0 df 20 91 41 b7 89 83 d3 67 a0 fe 97 08 df 32 f5 b9 88 8f
```

```
      e5 9e de 4e 61 2c f6 bd b1 fb be e6 f9 ef fe 13 01 00 00 2e 00
      33 00 24 00 1d 00 20 ab 16 0e 03 51 0f a0 3f d5 bd 6e 7a 94 f4
      00 31 16 35 cd 69 87 2e a6 e4 8a 08 71 5e e3 f0 24 2e 00 2b 00
      02 03 04

   ciphertext (127 octets):  16 03 03 00 7a 02 00 00 76 03 03 32 a4
      2f 56 c8 b8 59 cc 5d 80 f2 7f 48 d0 f2 96 d3 a5 bb 8e 05 28 08
      11 14 de 8c e3 84 d7 e0 df 20 91 41 b7 89 83 d3 67 a0 fe 97 08
      df 32 f5 b9 88 8f e5 9e de 4e 61 2c f6 bd b1 fb be e6 f9 ef fe
      13 01 00 00 2e 00 33 00 24 00 1d 00 20 ab 16 0e 03 51 0f a0 3f
      d5 bd 6e 7a 94 f4 00 31 16 35 cd 69 87 2e a6 e4 8a 08 71 5e e3
      f0 24 2e 00 2b 00 02 03 04

{server}  send change_cipher_spec record:

   payload (1 octets):  01

   ciphertext (6 octets):  14 03 03 00 01 01

{server}  derive secret for handshake "tls13 derived":

   PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
      10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
      97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

{server}  extract secret "handshake":

   salt (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
      16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

   IKM (32 octets):  d6 ee 52 33 ce 08 89 3e a5 eb d5 0f 0d 8a 25 bf
      ed 5f fd 57 82 32 31 19 46 91 bd 89 2b 8f 9a 50
```

```
   secret (32 octets):  2e 91 52 b1 5c ec 8f 81 92 f3 d5 a0 72 08 ad
      48 a9 7b 4e 06 f2 b8 22 9d f6 7b 7d 47 3e a8 42 d3

{server}  derive secret "tls13 c hs traffic":

   PRK (32 octets):  2e 91 52 b1 5c ec 8f 81 92 f3 d5 a0 72 08 ad 48
      a9 7b 4e 06 f2 b8 22 9d f6 7b 7d 47 3e a8 42 d3

   hash (32 octets):  ef ee 6c 01 8a 0f a3 ac 4c 61 ac 11 9c c8 fd da
      17 5e b8 c4 bd 4d 11 98 53 59 ca 1a f3 33 87 0b

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 68 73 20 74 72
      61 66 66 69 63 20 ef ee 6c 01 8a 0f a3 ac 4c 61 ac 11 9c c8 fd
      da 17 5e b8 c4 bd 4d 11 98 53 59 ca 1a f3 33 87 0b

   output (32 octets):  1b 92 72 16 81 91 bc c8 5e 46 45 96 e1 0b 79
      b8 09 a4 f6 36 02 e4 ad a5 b4 f2 c9 c0 b2 4d 27 37

{server}  derive secret "tls13 s hs traffic":

   PRK (32 octets):  2e 91 52 b1 5c ec 8f 81 92 f3 d5 a0 72 08 ad 48
      a9 7b 4e 06 f2 b8 22 9d f6 7b 7d 47 3e a8 42 d3

   hash (32 octets):  ef ee 6c 01 8a 0f a3 ac 4c 61 ac 11 9c c8 fd da
      17 5e b8 c4 bd 4d 11 98 53 59 ca 1a f3 33 87 0b

   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 68 73 20 74 72
      61 66 66 69 63 20 ef ee 6c 01 8a 0f a3 ac 4c 61 ac 11 9c c8 fd
      da 17 5e b8 c4 bd 4d 11 98 53 59 ca 1a f3 33 87 0b

   output (32 octets):  50 56 0b ed 1e 47 38 91 2d 43 d3 15 99 e0 7d
      5e ad ea f2 6b 18 9e 7b 75 e9 87 6f 42 07 2f b0 33

{server}  derive secret for master "tls13 derived":

   PRK (32 octets):  2e 91 52 b1 5c ec 8f 81 92 f3 d5 a0 72 08 ad 48
      a9 7b 4e 06 f2 b8 22 9d f6 7b 7d 47 3e a8 42 d3

   hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
      27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55

   info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
      20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
      64 9b 93 4c a4 95 99 1b 78 52 b8 55

   output (32 octets):  ef 79 6e a9 37 7c f8 94 b0 52 52 2b 22 9f cd
      70 a1 d7 c3 a3 2d ca 6c f5 1d 62 95 04 ef 1e e1 25
```

```
   {server}  extract secret "master":

      salt (32 octets):  ef 79 6e a9 37 7c f8 94 b0 52 52 2b 22 9f cd 70
         a1 d7 c3 a3 2d ca 6c f5 1d 62 95 04 ef 1e e1 25

      IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      secret (32 octets):  63 7d 72 8c c3 81 21 92 85 68 0b 8a bd 98 9c
         a3 7a c7 36 68 0c cb 47 8a 0f 28 11 07 2a 89 88 19

   {server}  derive write traffic keys for handshake data:

      PRK (32 octets):  50 56 0b ed 1e 47 38 91 2d 43 d3 15 99 e0 7d 5e
         ad ea f2 6b 18 9e 7b 75 e9 87 6f 42 07 2f b0 33

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  7d cd 41 e1 40 51 3f be 6a f5 22 a4 da 7f
         57 5b

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  77 ee 98 da ae 5c 82 24 7d 30 40 7f

   {server}  send a EncryptedExtensions handshake message

   {server}  send a Certificate handshake message

   {server}  send a CertificateVerify handshake message

   {server}  calculate finished "tls13 finished":

      PRK (32 octets):  50 56 0b ed 1e 47 38 91 2d 43 d3 15 99 e0 7d 5e
         ad ea f2 6b 18 9e 7b 75 e9 87 6f 42 07 2f b0 33

      hash (0 octets):  (empty)

      info (18 octets):  00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65
         64 00

      output (32 octets):  d1 61 e3 34 21 df d7 05 aa 4c c8 bf a6 e4 4d
```

```
        42 c8 b2 5b f1 c6 e4 e7 b4 dc c6 cb de a9 c2 a3 a1

   {server}  send a Finished handshake message

   {server}  send handshake record:
```



```
      payload (657 octets):  08 00 00 24 00 22 00 0a 00 14 00 12 00 1d
         00 17 00 18 00 19 01 00 01 01 01 02 01 03 01 04 00 1c 00 02 40
         01 00 00 00 00 0b 00 01 b9 00 00 01 b5 00 01 b0 30 82 01 ac 30
         82 01 15 a0 03 02 01 02 02 01 02 30 0d 06 09 2a 86 48 86 f7 0d
         01 01 0b 05 00 30 0e 31 0c 30 0a 06 03 55 04 03 13 03 72 73 61
         30 1e 17 0d 31 36 30 37 33 30 30 31 32 33 35 39 5a 17 0d 32 36
         30 37 33 30 30 31 32 33 35 39 5a 30 0e 31 0c 30 0a 06 03 55 04
         03 13 03 72 73 61 30 81 9f 30 0d 06 09 2a 86 48 86 f7 0d 01 01
         01 05 00 03 81 8d 00 30 81 89 02 81 81 00 b4 bb 49 8f 82 79 30
         3d 98 08 36 39 9b 36 c6 98 8c 0c 68 de 55 e1 bd b8 26 d3 90 1a
         24 61 ea fd 2d e4 9a 91 d0 15 ab bc 9a 95 13 7a ce 6c 1a f1 9e
         aa 6a f9 8c 7c ed 43 12 09 98 e1 87 a8 0e e0 cc b0 52 4b 1b 01
         8c 3e 0b 63 26 4d 44 9a 6d 38 e2 2a 5f da 43 08 46 74 80 30 53
         0e f0 46 1c 8c a9 d9 ef bf ae 8e a6 d1 d0 3e 2b d1 93 ef f0 ab
         9a 80 02 c4 74 28 a6 d3 5a 8d 88 d7 9f 7f 1e 3f 02 03 01 00 01
         a3 1a 30 18 30 09 06 03 55 1d 13 04 02 30 00 30 0b 06 03 55 1d
         0f 04 04 03 02 05 a0 30 0d 06 09 2a 86 48 86 f7 0d 01 01 0b 05
         00 03 81 81 00 85 aa d2 a0 e5 b9 27 6b 90 8c 65 f7 3a 72 67 17
         06 18 a5 4c 5f 8a 7b 33 7d 2d f7 a5 94 36 54 17 f2 ea e8 f8 a5
         8c 8f 81 72 f9 31 9c f3 6b 7f d6 c5 5b 80 f2 1a 03 01 51 56 72
         60 96 fd 33 5e 5e 67 f2 db f1 02 70 2e 60 8c ca e6 be c1 fc 63
         a4 2a 99 be 5c 3e b7 10 7c 3c 54 e9 b9 eb 2b d5 20 3b 1c 3b 84
         e0 a8 b2 f7 59 40 9b a3 ea c9 d9 1d 40 2d cc 0c c8 f8 96 12 29
         ac 91 87 b4 2b 4d e1 00 00 0f 00 00 84 08 04 00 80 84 d9 e6 bb
         5f 60 86 63 13 c5 02 3b 34 5b b6 68 4a 63 6c 67 82 34 01 5d c8
         3b 80 3d 81 30 68 ba 48 03 e2 cc 26 7f f0 86 70 35 d4 b4 46 28
         64 4c 1e fb 90 82 0c 47 ce c2 14 23 98 c3 aa d3 cf 9d a6 2d d4
         c5 de 51 ac 82 0c 84 af 40 72 1b dd 67 bc 8b bd db 28 3b 75 14
         25 62 0c f5 b2 76 f2 32 c2 a0 5e 53 f1 6b 6a d6 cd cd a6 04 da
         f9 95 e6 f8 42 4a 1d fd 37 0c 58 d0 f7 b4 60 5f 1a 21 a9 14 00
         00 20 5b 6a d9 10 bc 48 94 47 7b 48 da 86 11 eb c4 de 20 25 72
         63 5f 9c 4a ac 81 a4 81 2e 82 bf c2 fd

      ciphertext (679 octets):  17 03 03 02 a2 28 cc 1b 2f 47 22 95 79
         9f 34 2e 49 90 56 09 07 73 a4 57 20 6f 79 a5 4b b8 ca 78 dc 42
```

```
      e7 54 e1 95 6d dd 1a 78 6e 4c e9 6f 8d a4 12 57 ce 53 17 b7 37
      60 7a c3 b6 f8 6d 6f 6d 1d 71 06 01 af c5 61 0c d8 fb 16 7c 6a
      29 99 1e 50 a6 f4 83 7f ff 89 c2 d0 66 58 01 de 54 6e c2 8c bf
      f1 d7 d5 c3 30 b0 60 48 4a 44 0c 54 1c b1 1f 58 88 4a 50 31 dd
      ae ac ac af ea 6c 34 5a 93 8b 8e ee 6a 57 10 68 05 79 52 a2 60
      f9 e4 d6 51 bc e2 d8 57 1c ec aa da 2d 9b 37 15 60 3f f4 77 dd
      3c cf bf e6 8f 3c 0c b1 4b 0f c0 60 e6 dc 3b 10 f0 1b 43 8f 22
      12 71 3a 4b 87 fb b1 0d fd 9c 5c 29 e7 8d bc 7f a6 03 89 94 0f
      4e 3e 17 d9 79 f1 45 73 4d 67 66 12 ee 25 c1 15 fc da 0d f5 2c
      d2 35 95 77 fc b1 c2 47 e8 bf 90 0e 7a 59 0c 7e 33 f5 ff 1b 0e
      d0 d2 90 35 b5 f7 77 df d2 0f 02 41 40 61 7e e3 2d 6f 5a 7f 1b
      09 4e 60 d1 b1 78 2e 73 ca 22 ae c2 5d 1d 5f d7 ac c8 f5 58 17
      df 92 fe 17 da 29 13 77 10 e7 aa 2e bb 7c a8 45 6b de 8a dd e7
```



```
      88 24 19 c1 b1 8d ba d9 a9 70 54 30 bd 94 71 86 53 f3 d2 fb 78
      2c 62 f1 7b ef c3 24 73 f4 ec 5c 5d 73 39 e6 32 1c 65 d7 a0 c8
      f3 c5 d5 c5 1b bb c3 a5 3d 16 60 c5 89 eb e5 dd 39 bd 1e 53 6f
      f3 ed 09 84 41 36 76 4a b8 8d 51 71 db 6f bd 32 81 ec e9 e5 96
      07 85 56 0a 6f 51 fc f6 63 e8 fc 82 bb 13 d1 9b 49 c4 56 bc c1
      16 32 6a 70 1f 22 3a 19 d4 a4 5d cc f6 87 b3 95 9a a0 36 dd f3
      58 30 98 87 4c d6 da 79 6f e1 29 26 c1 2a 2d 49 79 1b 2d 88 1f
      13 be c3 ec de b5 fb 69 50 b8 5a 36 14 13 7e ad 5e 26 9e 14 84
      ee 26 2b ba d4 b1 c9 cd 35 09 69 85 75 f8 90 19 a9 28 05 81 5a
      ef 89 91 f8 63 6e a7 d4 87 c1 1c 9c 4c cb aa 91 1c 6c 57 b5 bb
      28 29 95 b7 f9 c9 c7 33 3d 7d 8f b7 40 cd 5f 0b 55 85 cb 87 d8
      7c 91 4d 02 c0 f5 6a 93 88 73 03 b2 93 38 6b 8e fb 26 48 b4 e1
      10 be 9e bc f5 c0 76 92 41 79 da b2 b1 bc a2 ad 05 21 44 fa 3b
      eb 38 5c 5c 28 f1 17 01 cc 78 3e 7c d8 f8 cc 92 b9 26 93 af 71
      28 2d ec 09 64 29 66 1d 75 f7 b8 3d 69 b4 39 be 1b f5 4e 74 da
      c8 3d e6 62 c5 93 15 15 bf ed 52 e4 cd 3c ce a8 de 9f b2 2a f9
      01 a3 40 af a3 3a 7b 06 d5 a5 fd e5 ce 1d 2b 7a 72 c7 e2 ee f5
      ff 46 25 d8 5b bb 99 5f 39 25 da d8 66 c6 5e
```

{server} derive secret "tls13 c ap traffic":

```
   PRK (32 octets):  63 7d 72 8c c3 81 21 92 85 68 0b 8a bd 98 9c a3
      7a c7 36 68 0c cb 47 8a 0f 28 11 07 2a 89 88 19

   hash (32 octets):  e4 72 ce 71 b4 9c c4 44 32 c4 09 f7 66 4b 84 a5
      9d 7a 68 3d 3d d2 da 22 7c 9b 98 42 3e a2 a1 45

   info (54 octets):  00 20 12 74 6c 73 31 33 20 63 20 61 70 20 74 72
```

```
      61 66 66 69 63 20 e4 72 ce 71 b4 9c c4 44 32 c4 09 f7 66 4b 84
      a5 9d 7a 68 3d 3d d2 da 22 7c 9b 98 42 3e a2 a1 45

   output (32 octets):  b3 59 c9 26 e6 22 56 e6 10 3e 70 fb bc f9 07
      cb 5e e7 56 20 f8 95 a8 b0 e8 c0 05 a4 df ff 75 6c

{server}  derive secret "tls13 s ap traffic":

   PRK (32 octets):  63 7d 72 8c c3 81 21 92 85 68 0b 8a bd 98 9c a3
      7a c7 36 68 0c cb 47 8a 0f 28 11 07 2a 89 88 19

   hash (32 octets):  e4 72 ce 71 b4 9c c4 44 32 c4 09 f7 66 4b 84 a5
      9d 7a 68 3d 3d d2 da 22 7c 9b 98 42 3e a2 a1 45

   info (54 octets):  00 20 12 74 6c 73 31 33 20 73 20 61 70 20 74 72
      61 66 66 69 63 20 e4 72 ce 71 b4 9c c4 44 32 c4 09 f7 66 4b 84
      a5 9d 7a 68 3d 3d d2 da 22 7c 9b 98 42 3e a2 a1 45

   output (32 octets):  7f 64 01 84 e5 99 d2 8e c8 18 84 1c ff 13 92
      30 d5 16 9f 16 3b 1f 52 70 12 a3 8e 5d b8 1f 7b 4e
```



```
{server}  derive secret "tls13 exp master":

   PRK (32 octets):  63 7d 72 8c c3 81 21 92 85 68 0b 8a bd 98 9c a3
      7a c7 36 68 0c cb 47 8a 0f 28 11 07 2a 89 88 19

   hash (32 octets):  e4 72 ce 71 b4 9c c4 44 32 c4 09 f7 66 4b 84 a5
      9d 7a 68 3d 3d d2 da 22 7c 9b 98 42 3e a2 a1 45

   info (52 octets):  00 20 10 74 6c 73 31 33 20 65 78 70 20 6d 61 73
      74 65 72 20 e4 72 ce 71 b4 9c c4 44 32 c4 09 f7 66 4b 84 a5 9d
      7a 68 3d 3d d2 da 22 7c 9b 98 42 3e a2 a1 45

   output (32 octets):  92 a0 34 07 bc bd c9 8d 26 ae 38 80 8b d6 f1
      0c d0 47 14 2e c7 ef ac b8 f3 08 9a 7e 3e 52 87 d6

{server}  derive write traffic keys for application data:

   PRK (32 octets):  7f 64 01 84 e5 99 d2 8e c8 18 84 1c ff 13 92 30
      d5 16 9f 16 3b 1f 52 70 12 a3 8e 5d b8 1f 7b 4e

   key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00
```

```
      key output (16 octets):  9a 33 b7 ff 19 01 80 b3 05 47 fe 9f e3 12
         74 09

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  a1 18 3b 47 0d 16 7f 63 62 8d 8b 32

   {server}  derive read traffic keys for handshake data:

      PRK (32 octets):  1b 92 72 16 81 91 bc c8 5e 46 45 96 e1 0b 79 b8
         09 a4 f6 36 02 e4 ad a5 b4 f2 c9 c0 b2 4d 27 37

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00

      key output (16 octets):  e7 37 b9 b1 2f 31 56 81 54 fd 6b f2 53 22
         ac 53

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  4a a7 80 6d 4f 81 d5 93 7b 99 3b 26

   {client}  extract secret "early":

      salt:  (absent)
```



```
      IKM (32 octets):  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
         00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

      secret (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c
         e2 10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

   {client}  derive secret for handshake "tls13 derived":

      PRK (32 octets):  33 ad 0a 1c 60 7e c0 3b 09 e6 cd 98 93 68 0c e2
         10 ad f3 00 aa 1f 26 60 e1 b2 2e 10 f1 70 f9 2a

      hash (32 octets):  e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24
         27 ae 41 e4 64 9b 93 4c a4 95 99 1b 78 52 b8 55
```

```
      info (49 octets):  00 20 0d 74 6c 73 31 33 20 64 65 72 69 76 65 64
         20 e3 b0 c4 42 98 fc 1c 14 9a fb f4 c8 99 6f b9 24 27 ae 41 e4
         64 9b 93 4c a4 95 99 1b 78 52 b8 55

      output (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6
         97 16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

   {client}  extract secret "handshake":

      salt (32 octets):  6f 26 15 a1 08 c7 02 c5 67 8f 54 fc 9d ba b6 97
         16 c0 76 18 9c 48 25 0c eb ea c3 57 6c 36 11 ba

      IKM (32 octets):  d6 ee 52 33 ce 08 89 3e a5 eb d5 0f 0d 8a 25 bf
         ed 5f fd 57 82 32 31 19 46 91 bd 89 2b 8f 9a 50

      secret (32 octets):  2e 91 52 b1 5c ec 8f 81 92 f3 d5 a0 72 08 ad
         48 a9 7b 4e 06 f2 b8 22 9d f6 7b 7d 47 3e a8 42 d3

   {client}  derive secret "tls13 c hs traffic" (same as server)

   {client}  derive secret "tls13 s hs traffic" (same as server)

   {client}  derive secret for master "tls13 derived" (same as server)

   {client}  extract secret "master" (same as server)

   {client}  derive read traffic keys for handshake data:

      PRK (32 octets):  50 56 0b ed 1e 47 38 91 2d 43 d3 15 99 e0 7d 5e
         ad ea f2 6b 18 9e 7b 75 e9 87 6f 42 07 2f b0 33

      key info (13 octets):  00 10 09 74 6c 73 31 33 20 6b 65 79 00
```



```
      key output (16 octets):  7d cd 41 e1 40 51 3f be 6a f5 22 a4 da 7f
         57 5b

      iv info (12 octets):  00 0c 08 74 6c 73 31 33 20 69 76 00

      iv output (12 octets):  77 ee 98 da ae 5c 82 24 7d 30 40 7f
```

```
{client}  calculate finished "tls13 finished" (same as server)

{client}  derive secret "tls13 c ap traffic" (same as server)

{client}  derive secret "tls13 s ap traffic" (same as server)

{client}  derive secret "tls13 exp master" (same as server)

{client}  send change_cipher_spec record:

   payload (1 octets):  01

   ciphertext (6 octets):  14 03 03 00 01 01

{client}  derive write traffic keys for handshake data (same as
   server read traffic keys)

{client}  derive read traffic keys for application data (same as
   server write traffic keys)

{client}  calculate finished "tls13 finished":

   PRK (32 octets):  1b 92 72 16 81 91 bc c8 5e 46 45 96 e1 0b 79 b8
      09 a4 f6 36 02 e4 ad a5 b4 f2 c9 c0 b2 4d 27 37

   hash (0 octets):  (empty)

   info (18 octets):  00 20 0e 74 6c 73 31 33 20 66 69 6e 69 73 68 65
      64 00

   output (32 octets):  89 90 6b c2 96 20 2c dc 3c 10 2a 87 ff fe 99
      cc cd b9 2c b1 94 d2 7a 8b 2b 21 10 e6 8b 41 0c 78

{client}  send a Finished handshake message

{client}  send handshake record:

   payload (36 octets):  14 00 00 20 ed 87 35 55 93 d3 ef 08 33 0b 32
      69 13 0f e9 5f cd e6 3e 60 1d b1 85 88 35 e5 5b 45 c4 08 e5 c5
```

ciphertext (58 octets): 17 03 03 00 35 9a b0 af 58 6e 95 81 22 3d
c2 bb 71 4d 5b e3 9f c2 eb 04 31 35 84 82 25 23 6d 39 24 71 5e
f9 10 bc 81 4c 59 f6 d8 5a d2 a9 22 d5 c4 18 ba bc 48 fb 6b 3a
bc 5e

{client} derive write traffic keys for application data:

PRK (32 octets): b3 59 c9 26 e6 22 56 e6 10 3e 70 fb bc f9 07 cb
5e e7 56 20 f8 95 a8 b0 e8 c0 05 a4 df ff 75 6c

key info (13 octets): 00 10 09 74 6c 73 31 33 20 6b 65 79 00

key output (16 octets): de ef 7b 47 f8 c6 cd d2 dc 85 7a cf 80 a4
67 5d

iv info (12 octets): 00 0c 08 74 6c 73 31 33 20 69 76 00

iv output (12 octets): af b0 ec 8b 9a d9 04 61 f1 ec 04 b2

{client} derive secret "tls13 res master":

PRK (32 octets): 63 7d 72 8c c3 81 21 92 85 68 0b 8a bd 98 9c a3
7a c7 36 68 0c cb 47 8a 0f 28 11 07 2a 89 88 19

hash (32 octets): e8 2a 79 f7 32 a4 90 44 12 3b 22 ce f3 54 68 fb
db ab 49 f4 b3 a3 ae 5c 5d 34 e0 f1 12 a3 7c 01

info (52 octets): 00 20 10 74 6c 73 31 33 20 72 65 73 20 6d 61 73
74 65 72 20 e8 2a 79 f7 32 a4 90 44 12 3b 22 ce f3 54 68 fb db
ab 49 f4 b3 a3 ae 5c 5d 34 e0 f1 12 a3 7c 01

output (32 octets): 87 33 e8 d1 4e b4 de f0 0b bb e3 f1 65 92 68
73 44 5f 2b c0 23 3d e0 98 2b 59 35 ec 89 ca 50 78

{server} calculate finished "tls13 finished" (same as client)

{server} derive read traffic keys for application data (same as
client write traffic keys)

{server} derive secret "tls13 res master" (same as client)

{client} send alert record:

payload (2 octets): 01 00

ciphertext (24 octets): 17 03 03 00 13 5e 7e 60 d9 38 04 1b 9a fd
34 c2 ad ef 72 cb 00 a8 63 43

```
{server}  send alert record:

   payload (2 octets):  01 00

   ciphertext (24 octets):  17 03 03 00 13 f8 11 03 38 e0 0b 60 4c f8
      82 5f 93 d6 10 ee af 43 91 f8
```

8. Security Considerations

It probably isn't a good idea to use the private key here. If it weren't for the fact that it is too small to provide any meaningful security, it is now very well known.

9. IANA Considerations

This document makes no requests of IANA.

10. References

10.1. Normative References

[TLS13] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", draft-ietf-tls-tls13-28 (work in progress), March 2018.

10.2. Informative References

[FIPS186] National Institute of Standards and Technology (NIST), "Digital Signature Standard (DSS)", NIST PUB 186-4 , July 2013.

[RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)", RFC 5869, DOI 10.17487/RFC5869, May 2010, <https://www.rfc-editor.org/info/rfc5869>.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

10.3. URIs

[1] https://developer.mozilla.org/en-US/docs/Mozilla/Projects/NSS

Appendix A. Acknowledgements

This draft is generated using tests that were written for NSS [1]. None of this would have been possible without Franziskus Kiefer, Eric Rescorla and Tim Taubert, who did a lot of the work in NSS.

Author's Address

Martin Thomson
Mozilla

Email: martin.thomson@gmail.com