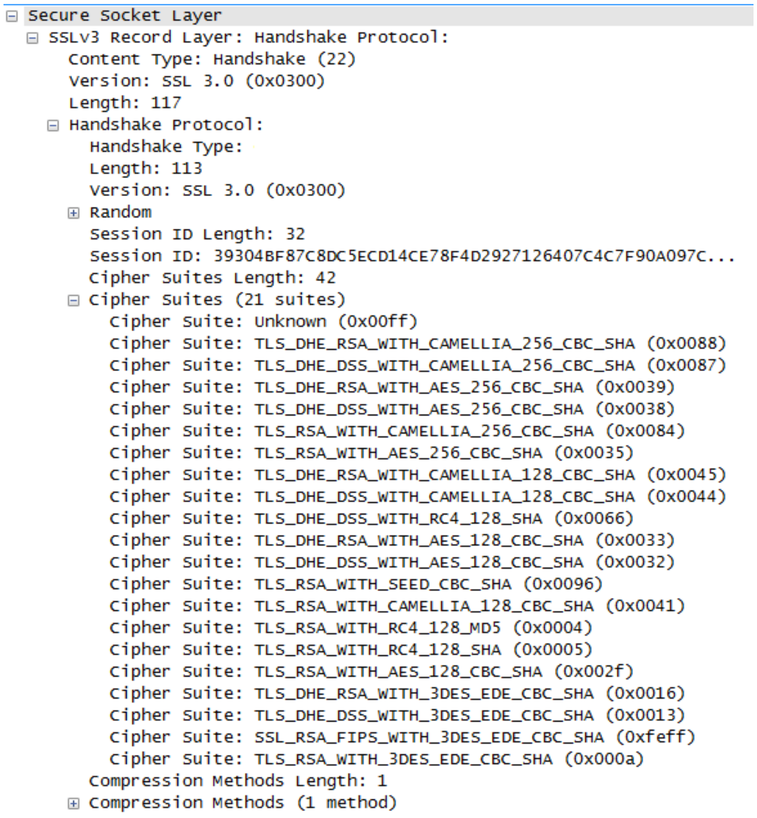
Internet Infrastructure Security (COMP444)
A6

Due at 11:55pm on 16 April 2015
Submission site: `https://submit.comp.polyu.edu.hk/`

Rocky K. C. Chang

April 9, 2015

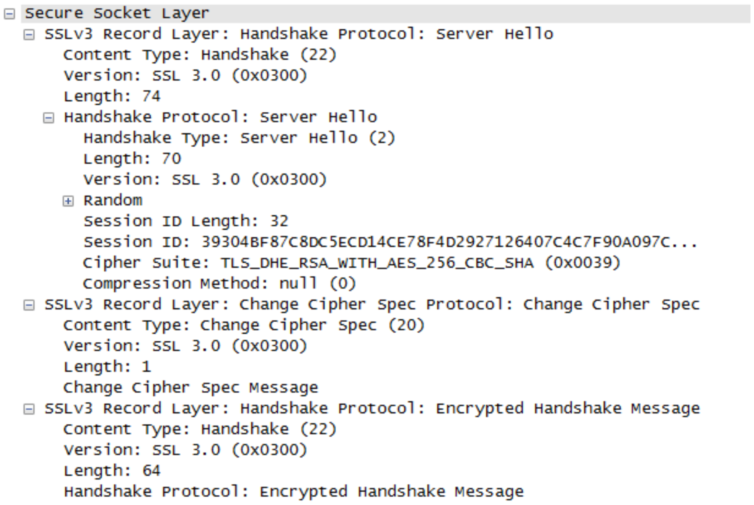
1. `[6 marks]` Figure 1 shows a TLS message.
   (a) (3 marks) What is this TLS message? Give evidence to support your answer.
   (b) (3 marks) What are the purposes of this TLS message?



```
⊟ Secure Socket Layer
  ⊟ SSLv3 Record Layer: Handshake Protocol:
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 117
    ⊟ Handshake Protocol:
        Handshake Type:
        Length: 113
        Version: SSL 3.0 (0x0300)
      ⊞ Random
        Session ID Length: 32
        Session ID: 39304BF87C8DC5ECD14CE78F4D2927126407C4C7F90A097C...
        Cipher Suites Length: 42
      ⊟ Cipher Suites (21 suites)
          Cipher Suite: Unknown (0x00ff)
          Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0088)
          Cipher Suite: TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA (0x0087)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
          Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
          Cipher Suite: TLS_RSA_WITH_CAMELLIA_256_CBC_SHA (0x0084)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
          Cipher Suite: TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0045)
          Cipher Suite: TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA (0x0044)
          Cipher Suite: TLS_DHE_DSS_WITH_RC4_128_SHA (0x0066)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
          Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
          Cipher Suite: TLS_RSA_WITH_SEED_CBC_SHA (0x0096)
          Cipher Suite: TLS_RSA_WITH_CAMELLIA_128_CBC_SHA (0x0041)
          Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
          Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
          Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
          Cipher Suite: SSL_RSA_FIPS_WITH_3DES_EDE_CBC_SHA (0xfeff)
          Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
        Compression Methods Length: 1
      ⊞ Compression Methods (1 method)
```

Figure 1: A TLS message.

2. `[6 marks]` Figure 2 shows a TLS message sent by a server. Does this TLS session use session reuse? Give sufficient evidence to support your answer.



```
⊟ Secure Socket Layer
  ⊟ SSLv3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 74
    ⊟ Handshake Protocol: Server Hello
        Handshake Type: Server Hello (2)
        Length: 70
        Version: SSL 3.0 (0x0300)
      ⊞ Random
        Session ID Length: 32
        Session ID: 39304BF87C8DC5ECD14CE78F4D2927126407C4C7F90A097C...
        Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
        Compression Method: null (0)
  ⊟ SSLv3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: SSL 3.0 (0x0300)
      Length: 1
      Change Cipher Spec Message
  ⊟ SSLv3 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 64
      Handshake Protocol: Encrypted Handshake Message
```

Figure 2: A TLS message sent by server.

3. `[16 marks]` (A speedy SSL) Recall that a SSL client can request to resume a SSL session. The session can be resumed only if the session states are still kept by the server. However, a busy server may not be able to keep the session states. A more logical approach is to keep those states on the client side. In this question we are exploring such possibility.

   The idea is that after a full handshake with a SSL server, a client may cache some of the states that most likely would not change in the next handshake with the server. There are 2 kinds of such states:

   (a) States dependent on the server's configurations (assume no client authentication)

      i. The server's certificate chain, and

      ii. The server's Diffie-Hellman group (if any)

   (b) States dependent on the interaction between the server and client

      i. The preferred client-server cipher suite; and

      ii. The preferred client-server compression method.

   Now consider the following handshake message exchange for a client to establish a SSL connection using the states kept in the previous handshaking session. Same as the lecture slides, we assume server authentication using RSA.

```
Client                                                 Server

ClientHello + speedy-hash
ClientKeyExchange
                             -------->
                                                    ServerHello
                                             [ChangeCipherSpec]
                                                       Finished
                             <--------
[ChangeCipherSpec]
```

```
Finished
                             -------->

Application Data             <------->        Application Data
```

The speedy-hash is the hash value of the states obtained from the previous SSL session.

(a) [3 marks] What is the purpose of sending speedy-hash with the ClientHello message?

(b) [3 marks] Explain why the client can send the ClientKeyExchange message immediately after the ClientHello message.

(c) [4 marks] Explain why the server can send all the handshaking messages in 1 “flow”? A flow is an uninterrupted sequence of messages sent from one side to the other.

(d) [3 marks] How does much this new handshaking speed up the ordinary one?

(e) [3 marks] Consider that the client includes a nonzero session ID in the ClientHello message in the new handshaking. If the server chooses to resume a session instead of using the cached states on the client side, draw the resulting protocol exchange and comment on the difference(s) between this exchange with the ordinary session reuse.