ICS 35.020
L 80

中华人民共和国国家标准

GB/T 28457—2012

SSL 协议应用测试规范

Testing specification for applications of SSL protocol

2012-06-29 发布　　2012-10-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　　次

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由全国信息安全标准化技术委员会(SAC/TC 260)提出并归口。

本标准的起草单位:信息工程大学信息工程学院。

本标准的主要起草人:颜学雄、王清贤、曾勇军、刘琰、耿俊燕、尹中旭。

引　言

目前，市场上 SSL 协议应用相关产品比较多，为了保证相关产品的质量，需对其进行测试。为统一产品的开发方、第三方授权测试认证单位和产品用户方对 SSL 协议应用相关产品的测试活动，并便于实现测试结果的相互认可和可重复性，特制定本标准。本标准规范了 SSL 协议应用相关产品的测试内容和基本测试过程。

为消除自然语言描述带来的二义性，本标准以有限状态自动机理论为基础，建立了 SSL 协议规范的形式化模型，并由此模型生成 SSL 协议应用的功能测试规范。相应的形式化模型见附录 A。

目前，还没有标准化的 SSL 协议应用测试工具。本标准没有规范具体的测试工具和测试环境，但为了规范测试人员的测试活动，本标准给出了测试指南，旨在规范测试基本步骤和关键要点，测试人员可以在此基础上，选择相关的辅助工具，产生具体的测试用例，并进行测试。

SSL 协议应用测试规范

1 范围

本标准规定了 SSL 协议应用的测试内容和基本测试步骤。

本标准适用于 SSL 协议应用的开发单位、第三方授权测试认证机构、用户等对 SSL 协议应用的测试。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 5271.8—2001 信息技术 词汇 第 8 部分：安全

GB/T 17178.1—1997 信息技术 开放系统互连 一致性测试方法和框架 第 1 部分：基本概念

3 术语和定义

GB/T 5271.8 界定的以及下列术语和定义适用于本文件。

3.1

SSL 协议 Secure Sockets Layer Protocol

一种应用于传输层的安全协议，用于构建客户端和服务器之间的安全通道，它提供保密性、完整性和可选的身份鉴别等安全功能。

3.2

SSL 协议应用 application of the SSL protocol

按照 SSL 协议规范标准实现的产品或功能模块。

3.3

客户端 client

主动发出 SSL 协议连接请求的通信方。

3.4

服务器 server

接收 SSL 协议连接请求的通信方。

3.5

功能测试 function testing

测试 SSL 协议应用的基本功能，包括基本连通测试、安全功能测试和行为测试。

3.6

性能测试 performance testing

测试在不同的网络负载情况下 SSL 协议应用的性能参数。

3.7

互操作性测试 interoperability testing

测试不同 SSL 协议应用之间的互操作能力。

3.8

健壮性　robustness

健壮性是描述一个系统或者一个组件在无效数据输入或者在高强度输入等环境下，其各项功能可保持正确运行的程度。

3.9

健壮性测试　robustness testing

测试 SSL 协议应用对错误(包括无效的、非期望输入或者恶意攻击)的有效处理能力。

3.10

基本连通测试　basic interconnection testing

测试 SSL 协议应用的基本连通的情况。

3.11

行为测试　behaviour testing

测试 SSL 协议应用的动态行为的符合性，包括正确输入行为符合性测试和非正确输入行为符合性测试。

3.12

连接建立时延　connection established delay

准备数据通信前的协商过程所需要的时间。

3.13

传送时延　transmission delay

数据从一方传送到另一方所需要的时间。

3.14

连接拆除时延　connection released delay

连接拆除所需要的时间。包括两种类型，一是客户端完成服务后，请求连接拆除的时延；二是服务器主动连接拆除的时延。

3.15

吞吐量　throughput

丢包率为零的情况下，单位时间内传输有效数据的数量。

3.16

剩余错误比率　residual error ratio

在给定的时间间隔内，传送不正确、丢失或者重复的数据量和传输正确的数据量之比。

3.17

失败概率　probability of failure

包括连接建立失败概率、传送失败概率以及连接拆除失败概率等。

3.18

新建连接　new connections

SSL 协议服务器每秒钟能够建立的 SSL 连接数。

3.19

并发连接数　max connections

SSL 服务器能够同时处理的最大 SSL 连接数。

4　缩略语

SSL　　安全套接层协议　　Secure Sockets Layer

TLS	传输层安全协议	Transport Layer Security
MAC	消息鉴别码	Message Authentication Code
ID	会话标识	IDentity
CA	证书认证机构	Certificate Authority

5 测试相关说明

5.1 测试对象说明

SSL 协议应用是指按照 SSL 协议规范标准实现的产品或功能模块。本标准涉及的 SSL 协议版本包括 SSL3.0[1]、TLS1.0(SSL3.1)[2,3]和 TLS1.1[4,5]。

SSL 协议规范是以本标准涉及的 SSL 协议版本为基础，规范了 SSL 协议应用应符合的安全服务功能规范、SSL 会话模式规范和 SSL 消息格式规范，详细说明见附录 A SSL 协议规范说明。

5.2 测试内容说明

本标准规范 SSL 协议应用的测试内容，包括功能测试、性能测试、健壮性测试和互操作性测试。

功能测试包括基本连通测试、安全功能测试和行为测试。基本连通测试完成 SSL 协议应用的基本连通情况测试，它是后续其他测试的基础，如果本项测试失败，则其他测试无法进行；安全功能测试包括密码算法协商功能测试、身份鉴别功能测试、数据保密性保护功能测试和数据完整性保护功能测试，安全功能中涉及的密码算法应符合国家密码管理的相关规定和要求；为保证各项安全功能的有效性，SSL 协议应用的动态行为必须符合规范要求，行为测试完成 SSL 协议应用的动态行为的符合性测试，包括正确输入行为符合性测试和非正确输入符合性测试。

性能测试完成不同网络负载情况下，SSL 协议应用的性能参数测试，包括连接建立时间、传送时延、连接拆除时间、吞吐量、剩余错误比率、失败概率、新建连接数和并发连接数。

健壮性是描述一个系统或者组件在无效数据输入或高强度输入等环境下，其各项功能可保持正确运行的程度。SSL 协议应用属于安全产品，其健壮性对于其安全功能的有效性至关重要。健壮性测试完成 SSL 协议应用对错误的有效处理能力，包括无效报文处理能力、异常事件处理能力和高强度负载处理能力。

互操作性测试完成不同厂家 SSL 协议应用间的互操作能力测试，包括基本连通互操性作测试和安全功能互操作测试。基本连通互操作性测试内容和功能测试中的基本连通测试内容一样，安全功能互操作性测试和功能测试中的安全功能测试内容一样。它们之间的区别在于，功能测试是 SSL 协议应用和测试系统之间的交互，而互操作性测试是不同的 SSL 协议应用间的交互(见 5.3 测试环境说明的内容)，同时对于错误的判别方法有所不同。本标准中，对于互操作性测试的相关内容不重复描述，测试人员可参照相关内容完成互操作性测试。

5.3 测试环境说明

虽然 SSL 协议应用测试不依赖于特定的测试工具，但是为规范测试人员的测试活动，本标准推荐采用 GB/T 17178.1—1997 中的远程测试方法，并依照此方法给出了测试步骤指南。其目的在于规范测试基本步骤和关键要点，测试人员可在此基础上，选择相关的测试辅助工具，产生具体的测试用例，并进行测试。

SSL 协议应用测试基本环境如图 1 所示。

当进行功能测试和健壮性测试时，被测对象 1 或被测对象 2 可能是 SSL 协议应用的客户端或服务器，测试系统和被测对象进行消息交互，通过对消息的判别，完成各项功能测试。

当进行性能测试时候，被测对象 1 和被测对象 2，分别对应被测 SSL 协议应用的客户端和服务器，通过对客户端和服务器交互消息的判别和统计，完成各项性能测试。

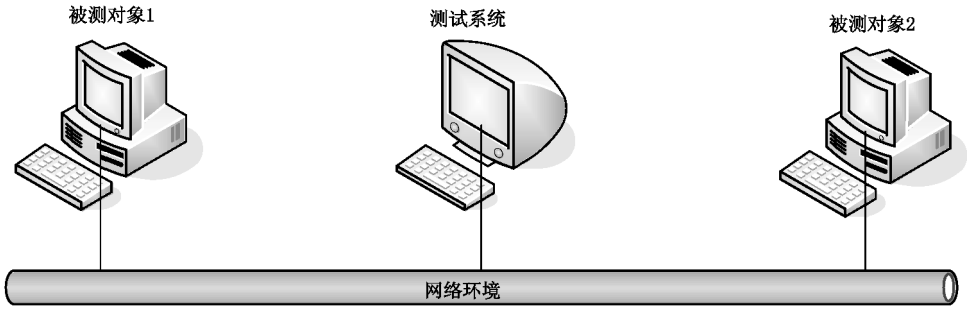


图 1　SSL 协议应用测试基本环境

当进行互操作性测试时，被测对象 1 和被测对象 2 分别是不同厂家实现的 SSL 协议产品的客户端和服务器，通过分析客户端和服务器的消息交互，完成各项互操作性测试。

6　测试内容规范

6.1　匿名会话模式测试内容

6.1.1　功能测试

6.1.1.1　基本连通测试

测试 SSL 协议应用在匿名会话模式下，客户端和服务器间的基本连通能力。

6.1.1.2　安全功能测试

测试内容如下：

a）　密码协商功能

1）　密码算法协商：协商会话两端共用的密钥交换算法、加密算法和 MAC 算法，这些算法必须在线完成协商；

2）　共享密钥协商：根据协商的密钥交换算法，会话两端完成共享密钥的协商；

b）　数据保密性保护功能：客户端和服务器能够应用协商好的加密算法以及共享密钥对传输的数据加密，以保护数据的保密性；

c）　数据完整性保护功能：客户端和服务器能够应用协商好的 MAC 算法以及共享密钥对传输的数据进行完整性保护。

6.1.1.3　客户端行为测试

在匿名会话模式下，客户端行为状态见图 A.2，其行为测试的内容包括：

a）　初始状态

在接收 HelloRequest * 消息的条件下，或者在未接收任何消息的条件下，发送 ClientHello 消息。HelloRequest 消息和 ClientHello 消息应符合规范要求。

除了接收 HelloRequest 消息外，忽略其他一切可能接收到的消息。

b）　握手状态

应依次接收 ServerHello 消息、ServerKeyExchange * 消息、ServerHelloDone 消息，中间不应有其他任何消息，消息格式应符合规范要求。然后，应依次发送 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收 ServerHello 消息、ServerKeyExchange * 消息、ServerHelloDone 消息，如果是其他的非警报协议消息或者其他的次序，则发送 unexpected_message 警报协议消息，并关闭当前连接。
2) illegal_parameter：接收到的 ServerHello 消息、ServerKeyExchange * 消息、ServerHelloDone 消息中的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。
3) record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
4) decode_error：在握手状态下，对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
5) protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
6) internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
7) 接收服务器警报消息：接收到服务器发送过来的警报协议消息，则关闭当前连接。

c) 密钥交换状态

应依次接收 ChangeCipherSpec 消息、服务器 Finished 消息，中间不应有其他消息，消息格式应符合规范要求。此时，客户端可发送应用数据消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收 ChangeCipherSpec 消息、服务器 Finished 消息，如果是其他的非警报协议消息或者其他的次序，则发送 unexpected_message 警报协议消息，并关闭当前连接。
2) bad_record_mac：处理服务器 Finished 消息时，MAC 验证失败，则发送 bad_record_mac 警报协议消息，并关闭当前连接。
3) decompression_failure：处理服务器 Finished 消息时，如果使用了压缩处理，并且消息解压缩失败，则发送 decompression_failure 警报协议消息，并关闭当前连接。
4) illegal_parameter：接收到的 ChangeCipherSpec 消息、服务器 Finished 消息中的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。
5) decryption_failed：处理服务器 Finished 消息时，解密出来的数据不符合密码算法的要求（如数据的长度不是某数的整数倍，或者填充的数据的值不符合要求），则发送 decryption_failed 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
6) record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
7) decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，

并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

8) decrypt_error:处理服务器 Finished 消息时,验证签名过程或者数据解密中出现错误,则发送 decrypt_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9) protocol_version:对接收到的任何消息,如果版本号不是规范中要求的,则发送 protocol_version 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10) internal_error:如果处理过程中出现了内部错误(如内存分配失败),则发送 internal_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

11) 接收服务器警报协议消息:接收到服务器发送过来的警报协议消息,则关闭当前连接。

d) 工作状态

应只有三种行为,一是接收或发送应用数据消息,消息格式应符合规范要求;二是接收服务器的关闭通知 Close_notify 警报协议消息,消息格式应符合规范要求;三是用户终止会话连接,客户端关闭连接,如果是 SSL3.0 版本,则同时向对方发送 Close_notify 警报协议消息,消息格式应符合规范要求,如果是 SSL3.1/TLS1.0 及以后版本,则同时向对方发送 user_canceled 警报协议消息,消息格式应符合规范要求。

如果是其他情况,则可发送有关的警报协议消息,消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下:

1) unexpected_message:应接收服务器应用数据消息或 Close_notify 消息,如果是其他的非警报协议消息,则发送 unexpected_message 警报协议消息,并关闭当前连接。

2) bad_record_mac:处理服务器应用数据消息时,MAC 验证失败,则发送 bad_record_mac 警报协议消息,并关闭当前连接。

3) decompression_failure:处理服务器应用数据消息时,如果使用了压缩处理,并且消息解压缩失败,则发送 decompression_failure 警报协议消息,并关闭当前连接。

4) decryption_failed:处理服务器应用数据消息时,解密出来的数据不符合密码算法的要求(数据的长度不是某数的整数倍,或者填充的数据的值不符合要求),则发送 decryption_failed 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

5) record_overflow:接收到的数据长度,超过了规定的长度,则发送 record_overflow 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

6) decode_error:对接收到的任何消息,如果解码失败,则发送 decode_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

7) decrypt_error:处理服务器应用数据消息时,验证签名过程或者数据解密中出现错误,则发送 decrypt_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

8) protocol_version:对接收到的任何消息,如果版本号不是规范中要求的,则发送 protocol_version 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9) internal_error:如果处理过程中出现了内部错误(如内存分配失败),则发送 internal_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10) 接收服务器警报协议消息:接收到服务器发送过来的警报协议消息,则关闭当前连接。

6.1.1.4 服务器行为测试

在匿名会话模式下，服务器行为状态见图 A.3，其行为测试的内容包括：

a) 初始状态

应接收 ClientHello 消息，消息格式应符合规范要求；然后，应依次发送 ServerHello 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应接收 ClientHello 消息，如果是其他的非警报协议消息，则发送 unexpected_message 警报协议消息，并关闭当前连接。
2) handshake_failure：如果无法完成安全参数的协商，则发送 handshake_failure 警报协议消息，并关闭当前连接。
3) illegal_parameter：接收到的客户端 ClientHello 消息的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。
4) record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
5) decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
6) protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
7) insufficient_security：如果服务器认为客户端 ClientHello 消息中的候选密码组安全强度不够时，则发送 insufficient_security 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
8) internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

b) 握手状态

应依次接收 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，消息格式应符合规范要求；然后，应依次发送 ChangeCipherSpec 消息和服务器 Finished 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，如果是其他的非警报协议消息，则发送 unexpected_message 警报协议消息，并关闭当前连接。
2) bad_record_mac：处理客户端发送过来的 Finished 消息时，MAC 验证失败，则发送 bad_record_mac 警报协议消息，并关闭当前连接。
3) decompression_failure：处理客户端发送过来的 Finished 消息时，如果使用了压缩处理，并且消息解压缩失败，则发送 decompression_failure 警报协议消息，并关闭当前连接。
4) illegal_parameter：接收到的任意消息的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。
5) decryption_failed：处理客户端 Finished 消息时，解密出来的数据不符合密码算法的要求

(如数据的长度不是某数的整数倍,或者填充的数据的值不符合要求),则发送 decryption_failed 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

6) record_overflow:接收到的数据长度,超过了规定的长度,则发送 record_overflow 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

7) decode_error:对接收到的任何消息,如果解码失败,则发送 decode_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

8) decrypt_error:处理客户端 Finished 消息时,验证签名过程或者数据解密中出现错误,则发送 decrypt_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9) protocol_version:对接收到的任何消息,如果版本号不是规范中要求的,则发送 protocol_version 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10) internal_error:如果处理过程中出现了内部错误(如内存分配失败),则发送 internal_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

11) 接收客户端警报消息:接收到客户端发送过来的警报协议消息,则关闭当前连接。

c) 工作状态

应只有三种行为,一是接收或发送应用数据消息,消息格式应符合规范要求;二是接收客户端的关闭通知 Close_notify 警报协议消息,消息格式应符合规范要求;三是用户终止会话连接,服务器关闭连接,如果是 SSL3.0 版本,则同时向对方发送 Close_notify 警报协议消息,消息格式应符合规范的要求,如果是 SSL3.1/TLS1.0 及以后版本,则同时向对方发送 user_canceled 警报协议消息,消息格式应符合规范要求。

如果是其他情况,则可发送有关的警报协议消息,消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下:

1) unexpected_message:应接收客户端应用数据消息或 Close_notify 消息,如果是其他的非警报协议消息,则发送 unexpected_message 警报协议消息,并关闭当前连接。

2) bad_record_mac:处理客户端应用数据消息时,MAC 验证失败,则发送 bad_record_mac 警报协议消息,并关闭当前连接。

3) decompression_failure:处理客户端应用数据消息时,如果使用了压缩处理,并且消息解压缩失败,则发送 decompression_failure 警报协议消息,并关闭当前连接。

4) decryption_failed:处理客户端应用数据消息时,解密出来的数据不符合密码算法的要求(如数据的长度不是某数的整数倍,或者填充的数据的值不符合要求),则发送 decryption_failed 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

5) record_overflow:接收到的数据长度,超过了规定的长度,则发送 record_overflow 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

6) decode_error:对接收到的任何消息,如果解码失败,则发送 decode_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

7) decrypt_error:处理客户端应用数据消息时,验证签名过程或者数据解密中出现错误,则发送 decrypt_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

8) protocol_version:对接收到的任何消息,如果版本号不是规范中要求的,则发送 protocol_

version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9) internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10) 接收客户端警报消息：接收到客户端发送过来的警报协议消息，则关闭当前连接。

6.1.2 性能测试

在匿名模式下，性能测试内容包括：

a) 连接建立时延；
b) 传送时延；
c) 连接拆除时延；
d) 吞吐量；
e) 剩余错误比率；
f) 失败概率；
g) 新建连接数；
h) 并发连接数。

6.1.3 健壮性测试

6.1.3.1 无效报文处理测试

测试内容包括：

a) 错误报文处理能力测试：当接收的报文消息头或者消息体中有错误时，SSL 协议应用客户端和服务器的处理能力测试；
b) 非期望报文处理能力测试：当接收到一个正确的报文，但报文不是当前状态所期望的输入报文时，SSL 协议应用客户端和服务器的处理能力测试。

6.1.3.2 异常事件处理测试

测试内容包括：

a) 连接中断处理能力测试：在连接突然非正常终止情况下，SSL 协议应用客户端和服务器的处理能力测试；
b) 等待超时处理测试能力：等待报文超时时，SSL 协议应用客户端和服务器的处理能力测试。

6.1.3.3 高强度负载测试

测试内容包括：

a) 网络高强度负载处理能力测试：在网络中流量负载比较大的情况下，SSL 协议应用客户端和服务器的处理能力测试；
b) 系统高强度负载处理能力测试：运行 SSL 协议应用客户端或者服务器的系统，多种资源消耗比较多的情况下（如 CPU 或者内存），SSL 协议应用客户端和服务器的处理能力测试。

6.1.4 互操作性测试

6.1.4.1 基本互连测试

在匿名会话模式下，测试不同 SSL 协议应用客户端和服务器之间的互连情况。

6.1.4.2 功能互操作性测试

在匿名会话模式下，应用不同 SSL 协议应用客户端和服务器互连后，其测试内容按 6.1.1.2 给出的要求。

6.2 服务器验证会话模式测试内容

6.2.1 功能测试

6.2.1.1 基本连通测试

测试 SSL 协议应用在服务器验证会话模式下，客户端和服务器间的基本连通能力。

6.2.1.2 安全功能测试

服务器验证会话模式在匿名会话模式提供的密码协商功能、数据保密性保护功能和数据完整性保护功能的基础上(按 6.1.1.2 给出的要求)，增加了服务器身份鉴别功能，具体如下：

a) 服务器身份证书递交功能：服务器应按照规范要求递交身份证书。

b) 服务器身份证书鉴别功能：客户端应能够对服务器递交的身份证书进行验证。

6.2.1.3 客户端行为测试

在服务器验证会话模式下，客户端行为状态见图 A.5，其行为测试的内容包括：

a) 初始状态

按 6.1.1.3 a)给出的要求。

b) 握手状态

应依次接收 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、ServerHelloDone 消息，中间不应有其他消息，消息格式应符合规范要求；然后，应依次发送 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，中间不应有其他消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、ServerHelloDone 消息，如果是其他的非警报协议消息或其他的次序，则发送 unexpected_message 警报协议消息，并关闭当前连接。

2) bad_certificate：如果验证服务器的 Certificate 消息中的身份证书失败，则发送 bad_certificate 警报协议消息，并关闭当前连接。

3) unsupported_certificate：如果服务器递交的身份证书格式，客户端本地不支持，则发送 unsupported_certificate 警报协议消息，并关闭当前连接。

4) certificate_revoked：如果服务器递交的身份证书已经被其颁发者撤消，则发送 certificate_revoked 警报协议消息，并关闭当前连接。

5) certificate_expired：如果服务器递交的身份证书已经过期，则发送 certificate_expired 警报协议消息，并关闭当前连接。

6) certificate_unknown：如果客户端处理服务器递交的身份证书过程中，出现异常，导致身份鉴别不成功，则发送 certificate_unknown 警报协议消息，并关闭当前连接。

7) illegal_parameter：接收到的 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、ServerHelloDone 消息中的任意一个域的值超过了规定的值，则发送 illegal_

parameter 警报协议消息，并关闭当前连接。

8） record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9） decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10） unknown_CA：客户端在处理服务器身份证书的时候，找不到信任 CA 的证书，则发送 unknown_CA 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

11） protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

12） internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

13） 接收服务器警报消息：接收到服务器发送过来的警报协议消息，则关闭当前连接。

c） 密钥交换状态

按 6.1.1.3 c）给出的要求。

d） 工作状态

按 6.1.1.3 d）给出的要求。

6.2.1.4 服务器行为测试

在服务器验证会话模式下，服务器行为状态见图 A.6，其行为测试的内容包括：

a） 初始状态

应接收 ClientHello 消息，消息格式应符合规范要求；然后，应依次发送 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1） unexpected_message：应接收 ClientHello 消息，如果是其他的非警报协议消息，则发送 unexpected_message 警报协议消息，并关闭当前连接。

2） handshake_failure：如果无法完成安全参数的协商，则发送 handshake_failure 警报协议消息，并关闭当前连接。

3） illegal_parameter：接收到 ClientHello 消息的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。

4） record_overflow：接收到的数据长度超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

5） decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

6） protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

7） insufficient_security：如果服务器认为 ClientHello 消息中的候选密码组安全强度不够时，则发送 insufficient_security 警报协议消息，并关闭当前连接。该情况处理方式出现在

SSL3.1/TLS1.0 及以后版本。

8) internal_error：如果处理过程中出现了内部错误(如内存分配失败)，则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

b) 握手状态

按 6.1.1.4 b)给出的要求。

c) 工作状态

按 6.1.1.4 c)给出的要求。

6.2.2 性能测试

按 6.1.2 给出的要求。

6.2.3 健壮性测试

按 6.1.3 给出的要求。

6.2.4 互操作性测试

6.2.4.1 基本互连测试

在服务器验证会话模式下，测试不同 SSL 协议应用客户端和服务器之间的互连情况。

6.2.4.2 功能互操作性测试

在服务器验证会话模式下，应用不同 SSL 协议应用客户端和服务器互连后，其测试内容按 6.2.1.2 给出的要求。

6.3 双方验证会话模式测试内容

6.3.1 功能测试

6.3.1.1 基本连通测试

测试 SSL 协议应用在双方验证会话模式下，客户端和服务器间的基本连通能力。

6.3.1.2 安全功能测试

双方验证会话模式在服务器验证会话模式提供的服务器身份鉴别功能、密码协商功能、数据保密性功能和数据完整性保护功能的基础上(按 6.2.1.2 给出的要求)，增加了客户端身份鉴别功能，具体如下：

a) 服务器请求客户端身份证书功能：服务器应按照规范要求主动请求客户端的身份证书；

b) 客户端身份证书递交功能：客户端应按照规范递交身份证书。同时，客户端根据身份证书格式的不同，可能还需要递交证书验证信息；

c) 客户端身份证书验证功能：服务器应对客户端递交的身份证书以及验证信息进行验证。

6.3.1.3 客户端行为测试

在双方验证会话模式下，客户端行为状态见图 A.8，其行为测试的内容包括：

a) 初始状态

按 6.1.1.3 a)给出的要求。

b) 握手状态

应依次接收 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息和 ServerHelloDone 消息，中间不应有其他任何消息，消息格式应符合规范要求；然后，应依次发送 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息，ChangeCipherSpec 消息和客户端 Finished 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式都应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息、ServerHelloDone 消息，如果是其他的非警报协议消息或者其他的次序，则发送 unexpected_message 警报协议消息，并关闭当前连接。

2) no_certificate：如果客户端无法提供证书信息，则发送 no_certificate 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

3) bad_certificate：如果验证服务器的 Certificate 消息中的身份证书失败，则发送 bad_certificate 警报协议消息，并关闭当前连接。

4) unsupported_certificate：如果服务器递交的身份证书格式，客户端本地不支持，则发送 unsupported_certificate 警报协议消息，并关闭当前连接。

5) certificate_revoked：如果服务器递交的身份证书已经被其颁发者撤消，则发送 certificate_revoked 警报协议消息，并关闭当前连接。

6) certificate_expired：如果服务器递交的身份证书已经过期，则发送 certificate_expired 警报协议消息，并关闭当前连接。

7) certificate_unknown：如果客户端处理服务器递交的身份证书过程中，出现异常，导致身份验证不成功，则发送 certificate_unknown 警报协议消息，并关闭当前连接。

8) illegal_parameter：接收到的 ServerHello 消息、ServerKeyExchange * 消息、ServerHelloDone 消息中的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。

9) record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10) decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

11) unknown_CA：客户端在处理服务器身份证书的时候，找不到信任 CA 的证书，则发送 unknown_CA 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

12) protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

13) internal_error：如果处理过程中出现了内部错误(如内存分配失败)，则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

14) 接收服务器警报消息：接收到服务器发送过来的警报协议消息，则关闭当前连接。

c) 密钥交换状态

按 6.1.1.3 c)给出的要求。

d) 工作状态

按 6.1.1.3 d)给出的要求。

6.3.1.4 服务器行为测试

在双方验证会话模式下，服务器行为状态见图 A.9，其行为测试的内容包括：

a) 初始状态

应接收 ClientHello 消息，消息格式应符合规范要求；然后，应依次发送 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应接收 ClientHello 消息，如果是其他的非警报协议消息，则发送 unexpected_message 警报协议消息，并关闭当前连接。
2) handshake_failure：如果无法完成安全参数的协商，则发送 handshake_failure 警报协议消息，并关闭当前连接。
3) illegal_parameter：接收到的 ClientHello 消息的任意一个域的值超过了规范规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。
4) record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
5) decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
6) protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
7) insufficient_security：如果服务器认为 ClientHello 消息中的候选密码组安全强度不够时，则发送 insufficient_security 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。
8) internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

b) 握手状态

应依次接收到客户端 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息，消息格式应符合规范要求；然后，依次发送 ChangeCipherSpec 消息和服务器 Finished 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收客户端 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息，如果是其他的非警报协议消息，则发送 unexpected_message 警报协议消息，并关闭当前连接。
2) bad_record_mac：处理客户端 Finished 消息时，MAC 验证失败，则发送 bad_record_mac 警报协议消息，并关闭当前连接。
3) bad_certificate：如果验证客户端 Certificate 消息中的身份证书失败，或验证 CertificateVerify * 消息失败，则发送 bad_certificate 警报协议消息，并关闭当前连接。
4) unsupported_certificate：如果客户端递交的身份证书格式，服务器本地不支持，则发送 unsupported_certificate 警报协议消息，并关闭当前连接。

5） certificate_revoked：如果客户端递交的身份证书已经被其颁发者撤消，则发送 certificate_revoked 警报协议消息，并关闭当前连接。

6） certificate_expired：如果客户端递交的身份证书已经过期，则发送 certificate_expired 警报协议消息，并关闭当前连接。

7） certificate_unknown：如果服务器处理客户端递交的身份证书过程中，出现异常，导致身份验证不成功，则发送 certificate_unknown 警报协议消息，并关闭当前连接。

8） decompression_failure：处理客户端 Finished 消息时，如果使用了压缩处理，并且消息解压缩失败，则发送 decompression_failure 警报协议消息，并关闭当前连接。

9） illegal_parameter：接收到的任意消息的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。

10） decryption_failed：处理客户端 Finished 消息时，解密出来的数据不符合密码算法的要求（如数据的长度不是某数的整数倍，或者填充的数据的值不符合要求），则发送 decryption_failed 警报协议消息，并关闭当前连接。该情况处理方式出现在SSL3.1/TLS1.0 及以后版本。

11） record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

12） decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

13） decrypt_error：处理客户端 Finished 消息时，验证签名过程或者数据解密中出现错误，则发送 decrypt_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

14） protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

15） internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

16） 接收客户端警报消息：接收到客户端发送过来的警报协议消息，则关闭当前连接。

c） 工作状态

按 6.1.1.4 c）给出的要求。

6.3.2 性能测试

按 6.1.2 给出的要求。

6.3.3 健壮性测试

按 6.1.3 给出的要求。

6.3.4 互操作性测试

6.3.4.1 基本互连测试

在双方验证会话模式下，测试不同 SSL 协议应用客户端和服务器之间的互连情况。

6.3.4.2 功能互操作性测试

在双方验证会话模式下，应用不同 SSL 协议应用客户端和服务器互连后，其测试内容按 6.3.1.2 给出的要求。

6.4 重用会话模式测试内容

6.4.1 功能测试

6.4.1.1 安全功能测试

测试内容包括：

a) 重用会话协商功能：客户端和服务器之间宜能够启动重用会话协商，重用的会话是应用匿名会话模式、或服务器验证会话模式、或双方验证模式通过 SSL 握手协议建立的。

b) 数据安全保护功能：客户端和服务器能够应用协商好的安全功能保护通信数据，包括：

1) 数据保密性保护功能：应用重用会话的数据加密算法以及共享密钥，保护传输数据的保密性。

2) 数据完整性保护功能：应用重用会话的 MAC 算法以及共享密钥，保护传输数据的完整性。

6.4.1.2 客户端行为测试

在重用会话模式下，客户端行为状态见图 A.11，其行为测试的内容包括：

a) 初始状态

应在接收 HelloRequest * 消息或未接收任何消息的条件下，发送 ClientHello 消息，该消息中应指定重用会话 ID，消息格式应符合规范要求。

除了接收 HelloRequest 消息外，忽略其他一切可能接收到的消息。

b) 握手状态

应依次接收 ServerHello 消息、ChangeCipherSpec 消息和服务器 Finished 消息，中间不应有其他任何消息，消息格式应符合规范要求；然后，应依次发送 ChangeCipherSpec 消息和客户端 Finished 消息，消息格式应符合规范要求。

如果是其他情况，则可发送有关的警报协议消息，消息格式应符合规范要求。可能的处理情况和发送的警报协议消息如下：

1) unexpected_message：应依次接收 ServerHello 消息、ChangeCipherSpec 消息和服务器 Finished 消息，如果是其他的非警报协议消息或者其他的次序，则发送 unexpected_message 警报协议消息，并关闭当前连接。

2) bad_record_mac：处理服务器 Finished 消息时，MAC 验证失败，则发送 bad_record_mac 警报协议消息，并关闭当前连接。

3) decompression_failure：处理服务器 Finished 消息时，如果使用了压缩处理，并且消息解压缩失败，则发送 decompression_failure 警报协议消息，并关闭当前连接。

4) illegal_parameter：接收到的 ServerHello 消息、ChangeCipherSpec 消息、服务器 Finished 消息中的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。

5) decryption_failed：处理服务器 Finished 消息时，解密出来的数据不符合密码算法的要求（如数据的长度不是某数的整数倍，或者填充的数据的值不符合要求），则发送 decryption_failed 警报协议消息，并关闭当前连接。该情况处理方式出现在SSL3.1/TLS1.0 及以后版本。

6) record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

7) decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，

并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

8) decrypt_error:处理服务器 Finished 消息时,验证签名过程或者数据解密中出现错误,则发送 decrypt_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9) protocol_version:对接收到的任何消息,如果版本号不是规范中要求的,则发送 protocol_version 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10) internal_error:如果处理过程中出现了内部错误(如内存分配失败),则发送 internal_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

11) 接收服务器警报消息:接收到服务器发送过来的警报协议消息,则关闭当前连接。

c) 工作状态

按 6.1.1.3 d)给出的要求。

6.4.1.3 服务器行为测试

在重用会话模式下,服务器行为状态图见图 A.12,其行为测试的内容包括:

a) 初始状态

应接收 ClientHello 消息,消息格式应符合规范要求;然后,应依次发送 ServerHello 消息、ChangeCipherSpec 消息和服务器 Finished 消息,消息格式应符合规范要求。

如果是其他情况,则可能要发送有关的警报协议消息,消息格式应满足规范要求。可能的处理情况和发送的警报协议消息如下:

1) unexpected_message:应接收 ClientHello 消息,如果是其他的非警报协议消息,则发送 unexpected_message 警报协议消息,并关闭当前连接。

2) illegal_parameter:接收到的 ClientHello 消息的任意一个域的值超过了规范规定的值,则发送 illegal_parameter 警报协议消息,并关闭当前连接。

3) record_overflow:接收到的数据长度,超过了规定的长度,则发送 record_overflow 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

4) decode_error:对接收到的任何消息,如果解码失败,则发送 decode_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

5) protocol_version:对接收到的任何消息,如果版本号不是规范中要求的,则发送 protocol_version 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

6) internal_error:如果处理过程中出现了内部错误(如内存分配失败),则发送 internal_error 警报协议消息,并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

b) 握手状态

应依次接收 ChangeCipherSpec 消息和客户端 Finished 消息,消息格式应符合规范要求;然后,可发送应用数据消息,消息格式应符合规范要求。

如果是其他情况,则可发送有关的警报协议消息,消息格式应满足规范要求。可能的处理情况和发送的警报协议消息如下:

1) unexpected_message:应依次接收到 ChangeCipherSpec 消息和客户端 Finished 消息,如果是其他的非警报协议消息,则发送 unexpected_message 警报协议消息,并关闭当前连接。

2) bad_record_mac:处理客户端 Finished 消息时,MAC 验证失败,则发送 bad_record_mac

警报协议消息，并关闭当前连接。

3） decompression_failure：处理客户端 Finished 消息时，如果使用了压缩处理，并且消息解压缩失败，则发送 decompression_failure 警报协议消息，并关闭当前连接。

4） illegal_parameter：接收到的任意消息的任意一个域的值超过了规定的值，则发送 illegal_parameter 警报协议消息，并关闭当前连接。

5） decryption_failed：处理客户端 Finished 消息时，解密出来的数据不符合密码算法的要求（如数据的长度不是某数的整数倍，或者填充的数据的值不符合要求），则发送 decryption_failed 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

6） record_overflow：接收到的数据长度，超过了规定的长度，则发送 record_overflow 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

7） decode_error：对接收到的任何消息，如果解码失败，则发送 decode_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

8） decrypt_error：处理客户端 Finished 消息时，验证签名过程或者数据解密中出现错误，则发送 decrypt_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

9） protocol_version：对接收到的任何消息，如果版本号不是规范中要求的，则发送 protocol_version 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

10） internal_error：如果处理过程中出现了内部错误（如内存分配失败），则发送 internal_error 警报协议消息，并关闭当前连接。该情况处理方式出现在 SSL3.1/TLS1.0 及以后版本。

11） 接收客户端警报消息：接收到客户端发送过来的警报协议消息，则关闭当前连接。

c） 工作状态

按 6.1.1.4 c)给出的要求。

6.4.2 性能测试内容

按 6.1.2 给出的要求。

6.4.3 健壮性测试内容

按 6.1.3 给出的要求。

6.4.4 功能互操作性测试

在重用会话模式下，应用不同 SSL 协议应用客户端和服务器互连后，其测试内容按 6.4.1.2 给出的要求。

7 测试步骤指南

7.1 匿名会话模式测试步骤

7.1.1 功能测试

7.1.1.1 基本连通测试

步骤如下：

第一步：配置待测试的客户端和服务器到相应会话模式。

第二步：客户端和服务器建立连接，如果不成功，则在当前会话模式下，基本连通测试失败。

7.1.1.2 安全功能测试

a) 密码协商功能测试

1) 算法协商功能测试

第一步：查看 ClientHello 消息中是否有多个供服务器选择的密码组，若无，则没有密码算法协商功能。

第二步：查看 ServerHello 消息中是否有已经协商好的密码组，若无，则没有密码算法协商功能。

2) 共享密钥协商功能测试

第一步：查看 ClientHello 消息中是否有多个供服务器选择的密码组，若无，则没有共享密钥协商功能。

第二步：查看 ServerHello 消息中是否有已经协商好的密码组，若无，则没有共享密钥协商功能。

第三步：查看协商好的密码组，看是否有密钥交换算法，若无，则没有共享密钥协商功能。

第四步：根据密钥交换算法，查看是否有有关的 ServerKeyExchange 消息和 ClientKeyExchange 消息，若无，或者消息和密钥交换算法不相符，则没有共享密钥协商功能。

b) 数据保密性保护功能测试

第一步：查看客户端 Finished 消息和服务器 Finished 消息，如果数据没有加密，则没有数据保密性保护功能。

第二步：查看客户端和服务器之间的应用数据消息，如果没有加密，则没有数据保密性保护功能。

c) 数据完整性保护功能测试

第一步：查看客户端 Finished 消息和服务器 Finished 消息，如果没有 MAC 数据，则没有数据完整性保护功能。

第二步：查看客户端和服务器之间的应用数据消息，如果没有 MAC 数据，则没有数据完整性保护功能。

7.1.1.3 客户端行为测试

a) 初始状态

第一步：服务器向客户端发送 HelloRequest 消息，查看客户端的响应，如果不是 ClientHello 消息，则客户端行为不符合规范要求。

第二步：客户端在没有接收到 HelloRequest 消息的情况，主动发送 ClientHello 消息，如果不能够主动发送，则客户端行为不符合规范要求。

第三步：查看 ClientHello 消息是否符合规范要求，如果不符合，则客户端行为不符合规范要求。

b) 握手状态

正常情况处理能力测试步骤如下：

第一步：依次向客户端发送正常的 ServerHello 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，查看客户端在处理这些消息后，响应消息是否依次是 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，如果不是，则客户

端行为不符合规范要求。

第二步：查看 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

其他情况客户端行为测试包括：

1） unexpected_message

第一步：向客户端发送 ServerHello 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息以外的其他消息，客户端应响应 unexpected_message 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

2） illegal_parameter

第一步：向客户端发送的 ServerHello 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息中的某些字段的值超过规范要求，客户端应响应 illegal_parameter 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

3） record_overflow

第一步：向客户端发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），客户端应响应 record_overflow 警报协议消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

4） decode_error

第一步：向客户端发送一些错误编码的数据包，客户端应响应 decode_error 警报协议消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

5） protocol_version

第一步：向客户端发送错误协议版本号的消息，客户端应响应 protocol_version 警报协议消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

6） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

7） 接收服务器警报协议消息

向客户端发送警报协议消息，客户端应关闭当前连接，会话结束，如果不是，则客户端行为不符合规范要求。

c） 密钥交换状态

正常情况处理能力测试步骤如下：

第一步：依次向客户端发送正常的 ChangeCipherSpec 消息和服务器 Finished 消息，此时，客户端应响应应用数据消息或不响应任何消息，如果不是，则客户端行为不符合规范要求。

第二步：如果客户端响应应用数据消息，则要查看应用数据消息是否应用了协商好的加密算法加密数据，如果不是，则客户端行为不符合规范要求。

第三步：如果客户端响应应用数据消息，则要查看应用数据消息是否应用了协商好的 MAC 算

法保护数据完整性，如果不是，则客户端行为不符合规范要求。

其他情况客户端行为测试包括：

1） unexpected_message

第一步：向客户端发送 ChangeCipherSpec 消息和服务器 Finished 消息以外的其他消息，客户端应响应 unexpected_message 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

2） bad_record_mac

第一步：在正常的服务器 ChangeCipherSpec 消息后，向客户端发送包含一个错误 MAC 数据的服务器 Finished 消息，客户端应响应 bad_record_mac 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 bad_record_mac 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

3） decompression_failure

第一步：在正常的服务器 ChangeCipherSpec 消息后，向客户端发送一个错误数据压缩的服务器 Finished 消息，客户端应响应 decompression_failure 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decompression_failure 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

4） illegal_parameter

第一步：向客户端发送的 ChangeCipherSpec 消息和服务器 Finished 消息中的某些字段的值超过规范要求，客户端应响应 illegal_parameter 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

5） decryption_failed

第一步：在正常的服务器 ChangeCipherSpec 消息后，向客户端发送一个无效加密数据（数据长度不符合加密算法要求或填充值不符合的数据）的服务器 Finished 消息，客户端应响应 decryption_failed 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decryption_failed 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

6） record_overflow

第一步：向客户端发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），客户端应响应 record_overflow 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

7） decode_error

第一步：向客户端发送一些错误编码的数据包，客户端应响应 decode_error 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

8） decrypt_error

第一步：在正常的服务器 ChangeCipherSpec 消息后，向客户端发送一个错误加密的服务

器 Finished 消息，客户端应响应 decrypt_error 警报协议消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decrypt_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

9） protocol_version

第一步：向客户端发送错误协议版本号的消息，客户端应响应 protocol_version 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

10） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

11） 接收服务器警报协议消息

向客户端发送警报协议消息，客户端应关闭当前连接，会话结束，如果不是，则客户端行为不符合规范要求。

d） 工作状态

正常情况处理能力测试步骤如下：

第一步：使用客户端发送应用数据消息，应用数据消息应使用协商的加密算法加密数据。若无，则客户端行为不符合规范要求。

第二步：使用客户端发送应用数据消息，应用数据消息应使用协商的 MAC 算法保护数据。若无，则客户端行为不符合规范要求。

第三步：向客户端发送正常应用数据消息，客户端不应响应任何消息，如果出现其他消息，则客户端行为不符合规范要求。

第四步：向客户端发送 Close_notify 警报协议消息，客户端应正常关闭当前连接，如果不是，则客户端行为不符合规范要求。

第五步：用户终止 SSL 会话，客户端应能够正常关闭当前连接，并向服务器发送 Close_notify 警报协议消息，如果不是，则客户端行为不符合规范要求。

其他情况客户端行为测试包括：

1） unexpected_message

第一步：向客户端发送应用数据消息和 Close_notify 消息以外的其他消息，客户端应响应 unexpected_message 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

2） bad_record_mac

第一步：向客户端发送一个包含错误 MAC 数据的应用数据消息，客户端应响应 bad_record_mac 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 bad_record_mac 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

3） decompression_failure

第一步：向客户端发送一个错误数据压缩的应用数据消息，客户端应响应 decompression_failure 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decompression_failure 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

4） decryption_failed

第一步：向客户端发送一个无效加密数据（数据长度不符合加密算法要求或者填充值不符合的数据）的应用数据消息，客户端应响应 decryption_failed 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decryption_failed 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

5） record_overflow

第一步：向客户端发送超过规范要求的长度的数据包（2 048＋2^{14} 字节），客户端应响应 record_overflow 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

6） decode_error

第一步：向客户端发送一些错误编码的数据包，客户端应响应 decode_error 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

7） decrypt_error

第一步：向客户端发送一个错误加密的应用数据消息，客户端应响应 decrypt_error 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decrypt_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

8） protocol_version

第一步：向客户端发送错误协议版本号的消息，客户端应响应 protocol_version 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

9） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

10） 接收服务器警报协议消息

向客户端发送警报协议消息，客户端应关闭当前连接，会话结束，如果不是，则客户端行为不符合规范要求。

7.1.1.4 服务器行为测试

a） 初始状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的 ClientHello 消息，查看服务器响应消息是否依次是 ServerHello 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，如果不是，则服务器行为不符合规范。

第二步：查看 ServerHello 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息是否符合规范，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1） unexpected_message

第一步：向服务器发送 ClientHello 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2） handshake_failure

第一步：向服务器发送 ClientHello 消息，并且服务器不支持消息中包含的所有密码组，服务器应响应 handshake_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 handshake_failure 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3） illegal_parameter

第一步：向服务器发送的 ClientHello 消息中的某些字段的值超过规范要求，服务器应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息，是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4） record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5） decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6） protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7） insufficient_security

第一步：向服务器发送 ClientHello 消息，并且服务器认为消息中包含的所有密码组安全强度不够，服务器应响应 insufficient_security 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 insufficient_security 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

8） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

b） 握手状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，查看服务器响应的消息是否依次是 ChangeCipherSpec 消息和服务器 Finished 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看服务器响应的 ChangeCipherSpec 消息和服务器 Finished 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1) unexpected_message

第一步：向服务器发送 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2) bad_record_mac

第一步：在正常的 ClientKeyExchange 消息和 ChangeCipherSpec 消息后，向服务器发送一个包含错误 MAC 数据的客户端 Finished 消息，服务器应响应 bad_record_mac 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 bad_record_mac 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3) decompression_failure

第一步：在正常的 ClientKeyExchange 消息和 ChangeCipherSpec 消息后，向服务器发送一个错误数据压缩的客户端 Finished 消息，服务器应响应 decompression_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decompression_failure 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4) illegal_parameter

第一步：向服务器发送的 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息中的某些字段的值超过规范要求，服务器应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5) decryption_failed

第一步：在正常的 ClientKeyExchange 消息和 ChangeCipherSpec 消息后，向服务器发送一个无效加密数据（数据长度不符合加密算法要求或者填充值不符合的数据）的客户端 Finished 消息，服务器应响应 decryption_failed 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decryption_failed 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6) record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7) decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

8) decrypt_error

第一步：在正常的 ClientKeyExchange 消息和 ChangeCipherSpec 消息后，向服务器发送

一个错误加密的客户端 Finished 消息，服务器应响应 decrypt_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decrypt_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

9） protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

10） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

11） 接收客户端警报协议消息

向服务器发送警报协议消息，服务器应关闭当前连接，会话结束，如果不是，则服务器行为不符合规范要求。

c） 工作状态

正常情况服务器行为测试基本步骤：

第一步：使用服务器发送应用数据消息，应用数据消息应使用协商的加密算法加密数据。若无，则服务器行为不符合规范要求。

第二步：使用服务器发送应用数据消息，应用数据消息应使用协商的 MAC 算法保护数据。若无，则服务器行为不符合规范要求。

第三步：向服务器发送正常应用数据消息，服务器不应响应任何消息，如果出现其他消息，则服务器行为不符合规范要求。

第四步：向服务器发送 Close_notify 消息，服务器应正常关闭当前连接，如果不是，则服务器行为不符合规范要求。

第五步：用户终止 SSL 会话，服务器能够正常关闭当前连接，并向客户端发送 Close_notify 消息，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1） unexpected_message

第一步：向服务器发送应用数据消息和 Close_notify 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2） bad_record_mac

第一步：向服务器发送一个包含错误 MAC 数据的应用数据消息，服务器应响应 bad_record_mac 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 bad_record_mac 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3） decompression_failure

第一步：向服务器发送一个错误数据压缩的应用数据消息，服务器应响应 decompression_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decompression_failure 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4） decryption_failed

第一步：向服务器发送一个无效加密数据（数据长度不符合加密算法要求或者填充值不符合的数据）的应用数据消息，服务器应响应 decryption_failed 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decryption_failed 消息，是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5） record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（2 048＋2^{14} 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6） decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7） decrypt_error

第一步：向服务器发送一个错误加密的应用数据消息，服务器应响应 decrypt_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decrypt_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

8） protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

9） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

10） 接收客户端警报协议消息

向服务器发送警报协议消息，服务器应关闭当前连接，会话结束，如果不是，则服务器行为不符合规范要求。

7.1.2 性能测试

a） 连接建立时延

第一步：配置客户端和服务器到相应会话模式，启动服务器。

第二步：启动客户端并建立连接，统计连接建立时间。

b） 传送时延

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：发送应用数据，并统计传输时延。

c） 连接拆除时延

第一步：在相应会话模式下，客户端或者服务器主动关闭当前连接。

第二步：统计正常连接拆除时延。

d） 吞吐量

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：在客户端和服务器之间传输大量应用数据，统计传输时间和传输数据量，然后计算吞吐量。

e) 剩余错误比率

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：在客户端和服务器之间传输大量应用数据，统计出错数据量，然后计算剩余错误比率。

f) 失败概率

第一步：配置客户端和服务器到相应会话模式，并多次建立、传送数据，然后关闭连接。

第二步：统计在此过程中的连接建立失败概率、传送失效概率以及连接拆除失败概率。

g) 新建连接数

第一步：配置客户端和服务器到相应会话模式。

第二步：客户端向服务器发起大量连接请求，统计单位时间内新建立的连接数。

h) 并发连接数

第一步：配置服务器到相应会话模式。

第二步：多个客户端同时向服务器建立大量连接并发送通信数据，统计服务器能够处理的最大连接数。

7.1.3 健壮性测试

7.1.3.1 无效报文处理测试

a) 错误报文处理能力测试

第一步：在 SSL 相应会话模式下，向客户端和服务器的各个状态发送错误报文（报文头错误或者内容错误）。

第二步：查看客户端和服务器的处理情况，应正常关闭连接，并响应有关的警报协议消息。

b) 非期望报文处理能力测试

第一步：在 SSL 相应会话模式下，向客户端和服务器的各个状态发送非期望报文。

第二步：查看客户端和服务器的处理情况，应正常关闭连接，并响应 unexpected_message 警报协议消息。

7.1.3.2 异常事件处理测试

a) 连接中断处理能力测试

第一步：在 SSL 相应会话模式下，在客户端和服务器会话过程中的各个阶段，断开物理连接。

第二步：查看客户端和服务器的处理能力。

b) 等待超时处理测试能力

第一步：在 SSL 相应会话模式下，延长数据交互时间。

第二步：查看客户端和服务器的处理能力。

7.1.3.3 高强度负载测试

a) 网络高强度负载处理能力测试

第一步：在网络负载很大的情况下，建立 SSL 相应会话并进行数据通信。

第二步：查看客户端和服务器的处理能力。

b) 系统高强度负载处理能力测试

第一步：在客户端和（或）服务器运行系统高强度负载的情况下（高内存或者 CPU 消耗），建立 SSL 相应会话并进行数据通信。

第二步：查看客户端和服务器的处理能力。

7.1.4 互操作性测试

7.1.4.1 基本互连测试

见 7.1.1.1 的内容。

7.1.4.2 功能互操作性测试

见 7.1.1.2 的内容。

7.2 服务器验证会话模式测试步骤

7.2.1 功能测试

7.2.1.1 基本连通测试

步骤如下：

第一步：配置待测试的客户端和服务器到相应会话模式。

第二步：客户端和服务器建立连接，如果不成功，则在当前会话模式下，基本连通测试失败。

7.2.1.2 安全功能测试

a) 服务器身份鉴别功能测试

1) 服务器身份证书递交功能测试

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：查看服务器 Certificate 消息中是否有服务器身份证书，若无，则服务器没有身份证书递交功能。

2) 服务器身份证书鉴别功能测试

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：向客户端发送正常的 Certificate 消息，查看会话协商是否能够继续，如果对服务器响应有关的服务器证书警报协议消息，则没有服务器身份证书鉴别功能。

第三步：向客户端发送不正常的 Certificate 消息（如错误的证书、过期的证书等），查看客户端的响应，如果没有发送有关的服务器证书警报协议消息，则没有服务器身份证书鉴别功能。

b) 密码算法协商功能测试

见 7.1.1.2 a)的内容。

c) 数据保密性保护功能测试

见 7.1.1.2 b)的内容。

d) 数据完整性保护功能测试

见 7.1.1.2 c)的内容。

7.2.1.3 客户端行为测试

a) 初始状态

见 7.1.1.3 a)的内容。

b) 握手状态

正常情况处理能力测试步骤如下：

第一步：依次向客户端发送正常的 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，查看客户端在处理这些消息后，响应消息是否依次是 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 ClientKeyExchange 消息、ChangeCipherSpec 消息和客户端 Finished 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

其他情况客户端行为测试包括：

1） unexpected_message

第一步：向客户端发送 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息以外的其他消息，客户端应响应 unexpected_message 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

2） bad_certificate

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书信息有错误（签名受到破坏），客户端应响应 bad_certificate 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 bad_certificate 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

3） unsupported_certificate

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书格式客户端不支持，客户端应响应 unsupported_certificate 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 unsupported_certificate 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

4） certificate_revoked

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书已经被撤消，客户端应响应 certificate_revoked 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 certificate_revoked 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

5） certificate_expired

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书已经过期，客户端应响应 certificate_expired 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 certificate_expired 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

6） certificate_unknown

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

7） illegal_parameter

第一步：向客户端发送的 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息中的某些字段的值超过规范要求，客户端应响应 illegal_parameter 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

8） record_overflow

第一步：向客户端发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），客户端应响应

record_overflow 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

9） decode_error

第一步：向客户端发送一些错误编码的数据包，客户端应响应 decode_error 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

10） unknown_CA

第一步：向客户端发送 Certificate 消息，并且证书颁发者不包含在客户端证书颁发者列表中，客户端应响应 unknown_CA 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 unknown_CA 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

11） protocol_version

第一步：向客户端发送错误协议版本号的消息，客户端应响应 protocol_version 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

12） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

13） 接收服务器警报协议消息

向客户端发送警报协议消息，客户端应关闭当前连接，会话结束，如果不是，则客户端行为不符合规范要求。

c） 密钥交换状态

见 7.1.1.3 c)的内容。

d） 工作状态

见 7.1.1.3 d)的内容。

7.2.1.4 服务器行为测试

a） 初始状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的 ClientHello 消息，查看服务器响应消息是否依次是 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息和 ServerHelloDone 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1） unexpected_message

第一步：向服务器发送 ClientHello 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2） handshake_failure

第一步：向服务器发送 ClientHello 消息，并且服务器不支持消息中包含的所有密码组，服务器应响应 handshake_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 handshake_failure 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3) illegal_parameter

第一步：向服务器发送的 ClientHello 消息中的某些字段的值超过规范要求，服务器应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4) record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5) decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6) protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7) insufficient_security

第一步：向服务器发送 ClientHello 消息，并且服务器认为消息中包含的所有密码组安全强度不够，服务器应响应 insufficient_security 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 insufficient_security 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

8) internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

b) 握手状态

见 7.1.1.4 b)的内容。

c) 工作状态

见 7.1.1.4 c)的内容。

7.2.2 性能测试

见 7.1.2 的内容。

7.2.3 健壮性测试

见 7.1.3 的内容。

7.2.4 互操作性测试

7.2.4.1 基本互连测试

见 7.2.1.1 的内容。

7.2.4.2 功能互操作性测试

见 7.2.1.2 的内容。

7.3 双方验证会话模式测试步骤

7.3.1 功能测试

7.3.1.1 基本连通测试

步骤如下：

第一步：配置待测试的客户端和服务器到相应会话模式。

第二步：客户端和服务器建立连接，如果不成功，则在当前会话模式下，基本连通测试失败。

7.3.1.2 安全功能测试

a) 客户端身份鉴别功能测试

客户端身份鉴别功能测试包括：

1) 服务器请求客户端身份证书功能

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：查看服务器是否发出正确的 CertificateRequest 消息，若无，则服务器没有身份证书请求功能。

2) 客户端身份证书递交功能

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：查看是否有客户端 Certificate 消息，并查看消息中是否有客户端身份证书，若无，则客户端没有身份证书递交功能。

3) 客户端身份证书鉴别功能

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：向服务器发送正常的 Certificate 消息，查看会话协商是否能够继续，如果服务器响应有关的客户端证书警报协议消息，则没有客户端身份证书鉴别功能。

第三步：向服务器发送不正常的 Certificate 消息（如错误的证书、过期的证书等），查看服务器的响应，如果没有发送有关的客户端证书警报协议消息，则没有客户端身份证书鉴别功能。

b) 服务器身份鉴别功能测试

见 7.2.1.2 a)的内容。

c) 密码协商功能测试

见 7.1.1.2 a)的内容。

d) 数据保密性保护功能测试

见 7.1.1.2 b)的内容。

e) 数据完整性保护功能测试

见 7.1.1.2 c)的内容。

7.3.1.3 客户端行为测试

a) 初始状态

见 7.1.1.3 a)的内容。

b) 握手状态

正常情况处理能力测试步骤如下：

第一步：依次向客户端发送正常的 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息和 ServerHelloDone 消息，查看客户端在处理这些消息后，响应消息是否依次是 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

其他情况客户端行为测试包括：

1) unexpected_message

第一步：向客户端发送 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息和 ServerHelloDone 消息以外的其他消息，客户端应响应 unexpected_message 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

2) no_certificate

第一步：在 CertificateRequest 消息，请求客户端没有的一种证书，客户端应响应 no_certificate 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 no_certificate 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

3) bad_certificate

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书信息有错误(签名受到破坏)，客户端应响应 bad_certificate 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 bad_certificate 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

4) unsupported_certificate

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书格式客户端不支持，客户端应响应 unsupported_certificate 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 unsupported_certificate 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

5) certificate_revoked

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书已经被撤消，客户端应响应 certificate_revoked 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 certificate_revoked 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

6) certificate_expired

第一步：向客户端发送 Certificate 消息，并且消息中包含的证书已经过期，客户端应响应 certificate_expired 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 certificate_expired 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

7） certificate_unknown

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

8） illegal_parameter

第一步：向客户端发送的 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息和 ServerHelloDone 消息中的某些字段的值超过规范要求，客户端应响应 illegal_parameter 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

9） record_overflow

第一步：向客户端发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），客户端应响应 record_overflow 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

10） decode_error

第一步：向客户端发送一些错误编码的数据包，客户端应响应 decode_error 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

11） unknown_CA

第一步：向客户端发送 Certificate 消息，并且证书颁发者不包含在客户端证书颁发者列表中，客户端应响应 unknown_CA 消息。如果不是，则客户端行为不符合规范要求。

第二步：查看 unknown_CA 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

12） protocol_version

第一步：向客户端发送错误协议版本号的消息，客户端应响应 protocol_version 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

13） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

14） 接收服务器警报协议消息

向客户端发送警报协议消息，客户端应关闭当前连接，会话结束，如果不是，则客户端行为不符合规范要求。

c） 密钥交换状态

见 7.1.1.3 c)的内容。

d） 工作状态

见 7.1.1.3 d)的内容。

7.3.1.4 服务器行为测试

a） 初始状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的 ClientHello 消息，查看服务器响应消息是否依次是 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息和 ServerHelloDone 消息，如果不是，则服务器行为不符合规范。

第二步：查看 ServerHello 消息、Certificate 消息、ServerKeyExchange * 消息、CertificateRequest 消息和 ServerHelloDone 消息是否符合规范，如果不是，则服务器行为不符合规范。

其他情况服务器行为测试包括：

1) unexpected_message

第一步：向服务器发送 ClientHello 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2) handshake_failure

第一步：向服务器发送 ClientHello 消息，并且服务器不支持消息中包含的所有密码组，服务器应响应 handshake_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 handshake_failure 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3) illegal_parameter

第一步：向服务器发送的 ClientHello 消息中的某些字段的值超过规范要求，服务器应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4) record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5) decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6) protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7) insufficient_security

第一步：向服务器发送 ClientHello 消息，并且服务器认为消息中包含的所有密码组安全强度不够，服务器应响应 insufficient_security 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 insufficient_security 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

8） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

b） 握手状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的客户端 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息，查看服务器响应的消息是否依次是 ChangeCipherSpec 消息和服务器 Finished 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看服务器响应的 ChangeCipherSpec 消息和服务器 Finished 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1） unexpected_message

第一步：向服务器发送 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2） bad_record_mac

第一步：在正常的 Certificate 消息、ClientKeyExchange 消息、ChangeCipherSpec 消息后，向服务器发送一个包含错误 MAC 数据的客户端 Finished 消息，服务器应响应 bad_record_mac 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 bad_record_mac 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3） bad_certificate

第一步：向服务器发送 Certificate 消息，并且消息中包含的证书信息有错误（签名受到破坏），服务器应响应 bad_certificate 消息。如果不是，则服务器行为不符合规范要求。

第二步：查看 bad_certificate 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4） unsupported_certificate

第一步：向服务器发送 Certificate 消息，并且消息中包含的证书格式服务器不支持，服务器应响应 unsupported_certificate 消息。如果不是，则服务器行为不符合规范要求。

第二步：查看 unsupported_certificate 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5） certificate_revoked

第一步：向服务器发送 Certificate 消息，并且消息中包含的证书已经被撤消，服务器应响应 certificate_revoked 消息。如果不是，则服务器行为不符合规范要求。

第二步：查看 certificate_revoked 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6） certificate_expired

第一步：向服务器发送 Certificate 消息，并且消息中包含的证书已经过期，服务器应响应 certificate_expired 消息。如果不是，则服务器行为不符合规范要求。

第二步：查看 certificate_expired 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7) certificate_unknown

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

8) decompression_failure

第一步：在正常的 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息和 ChangeCipherSpec 消息后，向服务器发送一个错误数据压缩的客户端 Finished 消息，服务器应响应 decompression_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decompression_failure 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

9) illegal_parameter

第一步：向服务器发送的 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息、ChangeCipherSpec 消息和客户端 Finished 消息中的某些字段的值超过规范要求，客户端应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息，是否符合规范要求，如果不是，则服务器行为不符合规范要求。

10) decryption_failed

第一步：在正常的 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息和 ChangeCipherSpec 消息后，向服务器发送一个无效加密数据（数据长度不符合加密算法要求或者填充值不符合的数据）的客户端 Finished 消息，服务器应响应 decryption_failed 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decryption_failed 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

11) record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

12) decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

13) decrypt_error

第一步：在正常的 Certificate 消息、ClientKeyExchange 消息、CertificateVerify * 消息和 ChangeCipherSpec 消息后，向服务器发送一个错误加密的客户端 Finished 消息，服务器应响应 decrypt_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decrypt_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

14) protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

15) internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

16) 接收客户端警报协议消息

向服务器发送警报协议消息，服务器应关闭当前连接，会话结束，如果不是，则服务器行为不符合规范要求。

c) 工作状态

见 7.1.1.4 c)的内容。

7.3.2 性能测试

见 7.1.2 的内容。

7.3.3 健壮性测试

见 7.1.3 的内容。

7.3.4 互操作性测试

7.3.4.1 基本互连测试

见 7.3.1.1 的内容。

7.3.4.2 功能互操作性测试

见 7.3.1.2 的内容。

7.4 重用会话模式测试步骤

7.4.1 功能测试

7.4.1.1 安全功能测试

a) 重用会话协商功能测试

第一步：配置客户端和服务器到相应会话模式，并建立连接。

第二步：查看 ClientHello 消息中是否有重用会话请求，若无，则没有重用会话协商功能。

第三步：查看服务器对含有重用会话请求的 ClientHello 消息的响应，如果出错，则没有重用会话协商功能。

b) 数据安全保护功能测试

1) 数据保密性保护功能测试

第一步：配置客户端和服务器到相应会话模式，建立连接，并进行数据通信。

第二步：查看客户端 Finished 消息和服务器 Finished 消息，如果数据没有加密，则没有数据保密性保护功能。

第三步：查看客户端和服务器之间的应用数据消息，如果没有加密，则没有数据保密性保护功能。

2) 数据完整性保护功能测试

第一步：配置客户端和服务器到相应会话模式，建立连接，并进行数据通信。

第二步：查看客户端 Finished 消息和服务器 Finished 消息，如果没有 MAC 数据，则没有数据完整性保护功能。

第三步：查看客户端和服务器之间的应用数据消息，如果没有 MAC 数据，则没有数据完整性保护功能。

7.4.1.2 客户端行为测试

a) 初始状态

第一步：客户端主动发送 ClientHello 消息，如果不能够主动发送，则客户端行为不符合规范要求。

第二步：查看 ClientHello 消息是否符合规范要求，如果不符合，则客户端不符合规范要求。

b) 握手状态

正常情况处理能力测试步骤如下：

第一步：依次向客户端发送正常的 ServerHello 消息、ChangeCipherSpec 消息和服务器 Finished 消息，查看客户端响应消息是否依次是 ChangeCipherSpec 消息和客户端 Finished 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 ChangeCipherSpec 消息和客户端 Finished 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

其他情况客户端行为测试包括：

1) unexpected_message

第一步：向客户端发送 ServerHello 消息、ChangeCipherSpec 消息和服务器 Finished 消息以外的其他消息，客户端应响应 unexpected_message 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

2) bad_record_mac

第一步：在正常的 ServerHello 消息和 ChangeCipherSpec 消息后，向客户端发送一个包含错误 MAC 数据的服务器 Finished 消息，客户端应响应 bad_record_mac 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 bad_record_mac 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

3) decompression_failure

第一步：在正常的 ServerHello 消息和 ChangeCipherSpec 消息后，向客户端发送一个错误数据压缩的服务器 Finished 消息，客户端应响应 decompression_failure 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decompression_failure 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

4) illegal_parameter

第一步：向客户端发送的 ServerHello 消息、ChangeCipherSpec 消息和服务器 Finished 消息中的某些字段的值超过规范要求，客户端应响应 illegal_parameter 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

5) decryption_failed

第一步：在正常的 ServerHello 消息和 ChangeCipherSpec 消息后，发送一个无效加密数据（数据长度不符合加密算法要求或填充值不符合要求的数据）的服务器 Finished 消息，客户端应响应 decryption_failed 消息，如果不是，则客户端行为不符合规范要求。

第二步：查看 decryption_failed 消息是否符合规范要求，如果不是，则客户端行为不符合

规范要求。

6） record_overflow

第一步：向客户端发送超过规范要求的长度的数据包（2 048＋2^{14}字节），客户端应响应record_overflow消息，如果不是，则客户端行为不符合规范要求。

第二步：查看record_overflow消息是否符合规范要求，如果不是，则客户行为不符合规范要求。

7） decode_error

第一步：向客户端发送一些错误编码的数据包，客户端应响应decode_error消息，如果不是，则客户端行为不符合规范要求。

第二步：查看decode_error消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

8） decrypt_error

第一步：在正常的ServerHello消息和ChangeCipherSpec消息后，向客户端发送一个错误加密的服务器Finished消息，客户端应响应decrypt_error警报协议消息，如果不是，则客户端行为不符合规范要求。

第二步：查看decrypt_error消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

9） protocol_version

第一步：向客户端发送错误协议版本号的消息，客户端应响应protocol_version消息，如果不是，则客户端行为不符合规范要求。

第二步：查看protocol_version消息是否符合规范要求，如果不是，则客户端行为不符合规范要求。

10） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

11） 接收服务器警报协议消息

向客户端发送警报协议消息，客户端应关闭当前连接，会话结束，如果不是，则客户端行为不符合规范要求。

c） 工作状态

见7.1.1.3 d)的内容。

7.4.1.3 服务器行为测试

a） 初始状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的ClientHello消息（包含重用会话请求），查看服务器响应消息是否依次是ServerHello消息、ChangeCipherSpec消息和服务器Finished消息，如果不是，则服务器行为不符合规范要求。

第二步：查看ServerHello消息、ChangeCipherSpec消息和服务器Finished消息是否符合规范，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1） unexpected_message

第一步：向服务器发送ClientHello消息以外的其他消息，服务器应响应unexpected_message消息，如果不是，则服务器行为不符合规范要求。

第二步：查看unexpected_message消息是否符合规范要求，如果不是，则服务器行为不符

合规范要求。

2） illegal_parameter

第一步：向服务器发送的 ClientHello 消息中的某些字段的值超过规范要求，服务器应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3） record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4） decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5） protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6） internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

b） 握手状态

正常情况服务器行为测试步骤如下：

第一步：向服务器发送正常的 ChangeCipherSpec 消息和客户端 Finished 消息，此时，服务器应响应应用数据消息或者不响应任何消息，如果不是，则服务器行为不符合规范要求。

第二步：如果服务器响应应用数据消息，则查看应用数据消息是否应用了协商好的加密算法加密数据，如果不是，则服务器行为不符合规范要求。

第三步：如果服务器响应应用数据消息，则查看应用数据消息是否应用了协商好的 MAC 算法保护数据完整性，如果不是，则服务器行为不符合规范要求。

其他情况服务器行为测试包括：

1） unexpected_message

第一步：向服务器发送 ChangeCipherSpec 消息和客户端 Finished 消息以外的其他消息，服务器应响应 unexpected_message 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 unexpected_message 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

2） bad_record_mac

第一步：在正常的 ChangeCipherSpec 消息后，向服务器发送一个包含错误 MAC 数据的客户端 Finished 消息，服务器应响应 bad_record_mac 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 bad_record_mac 消息，是否符合规范要求，如果不是，则服务器行为不符合规范要求。

3) decompression_failure

第一步：在正常的 ChangeCipherSpec 消息后，向服务器发送一个错误数据压缩的客户端 Finished 消息，服务器应响应 decompression_failure 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decompression_failure 消息，是否符合规范要求，如果不是，则服务器行为不符合规范要求。

4) illegal_parameter

第一步：向服务器发送的 ChangeCipherSpec 消息和客户端 Finished 消息中的某些字段的值超过规范要求，服务器应响应 illegal_parameter 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 illegal_parameter 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

5) decryption_failed

第一步：在正常的 ChangeCipherSpec 消息后，向服务器发送一个无效加密数据（数据长度不符合加密算法要求或者填充值不符合的数据）的客户端 Finished 消息，服务器应响应 decryption_failed 警报协议消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decryption_failed 消息，是否符合规范要求，如果不是，则服务器行为不符合规范要求。

6) record_overflow

第一步：向服务器发送超过规范要求的长度的数据包（$2\ 048+2^{14}$ 字节），服务器应响应 record_overflow 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 record_overflow 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

7) decode_error

第一步：向服务器发送一些错误编码的数据包，服务器应响应 decode_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decode_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

8) decrypt_error

第一步：在正常的 ChangeCipherSpec 消息后，向服务器发送一个错误加密的客户端 Finished 消息，服务器应响应 decrypt_error 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 decrypt_error 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

9) protocol_version

第一步：向服务器发送错误协议版本号的消息，服务器应响应 protocol_version 消息，如果不是，则服务器行为不符合规范要求。

第二步：查看 protocol_version 消息是否符合规范要求，如果不是，则服务器行为不符合规范要求。

10) internal_error

此类情况一般不应出现。出现这种情况，说明系统本身有问题。

11) 接收客户端警报协议消息

向服务器发送警报协议消息，服务器应关闭当前连接，会话结束，如果不是，则服务器行为不符合规范要求。

7.4.2 性能测试

见 7.1.2 的内容。

7.4.3 健壮性测试

见 7.1.3 的内容。

7.4.4 功能互操作性测试

见 7.4.1.1 的内容。

附 录 A
（资料性附录）
SSL 协议规范说明

A.1 SSL 协议规范基本内容

本标准涉及的 SSL 协议版本包括 SSL3.0[1]、SSL3.1/TLS1.0[2,3]和 TLS1.1[4]，SSL 协议规范的基本内容包括：

a) SSL 协议安全功能：SSL 协议应用应实现密码协商、保密性保护、完整性保护等功能，宜实现身份鉴别功能；

b) SSL 会话模式规范：SSL 协议应用可实现的会话模式包括匿名会话模式、服务器验证会话模式、双方验证会话模式和重用会话模式；

c) SSL 消息格式规范：规范不同会话模式下，不同 SSL 协议版本下的消息格式。

A.2 SSL 协议安全功能

SSL 协议应用应实现的安全功能包括：

a) 密码协商：SSL 协议应用在通信过程中，应能够协商双方支持的密钥交换算法、数据加密算法和 MAC 算法，应能够根据密钥交换算法实现在线共享密钥的协商，同时，协商过程中所应用的密码算法和有关运算应符合国家的有关标准和规定；

b) 数据保密性保护：SSL 协议应用应保护通信数据的保密性，所应用的密码算法应符合国家的有关标准和规定；

c) 数据完整性保护：SSL 协议应用应保护通信数据的完整性，所应用的密码算法应符合国家的有关标准和规定。

SSL 协议应用宜实现的安全功能包括：

d) 身份鉴别：SSL 协议应用可采取以下三种形式实现身份鉴别功能：

1) 不对服务器和客户端进行身份鉴别；

2) 只对服务器进行身份鉴别；

3) 同时对服务器和客户端进行身份鉴别。

身份鉴别中所应用的密码算法和数字证书应符合国家的有关标准和规定。

A.3 SSL 协议会话模式

A.3.1 SSL 会话安全参数

SSL 会话中的安全功能的具体实现，体现在会话安全参数的实例化，这些安全参数包括：

a) 会话端：参与会话的是服务器，或是客户端。

b) 数据加密算法：包括数据加密的算法、算法类型、算法密钥长度、密钥材料长度。

c) MAC 算法：包括消息认证的算法、算法输出长度。

d) 压缩算法：包括算法的名称和其他有关信息。

e) 主密钥：服务器/客户端共享的密钥，用来生成各种算法密钥的材料。

f) 客户端随机数：客户端产生的一个 32 字节随机数。

g) 服务器随机数：服务器产生的一个 32 字节随机数。

对于一个具体的会话，服务器存放有一套安全参数实例，客户端也有一套安全参数实例，这两套安全参数除了会话端不一样外，其他应是一样的。安全参数中具体值的含义以及表示形式，和具体应用的密码算法有关，SSL 协议应用应符合国家密码有关标准和规定的要求。

A.3.2 SSL 会话模式分类

根据会话过程提供的安全功能的不同，SSL 会话分为如下几类模式：

a) 匿名会话模式：在该模式下，SSL 协议应用提供密码协商、数据保密性保护和数据完整性保护等安全功能，但不提供进行身份鉴别功能；

b) 服务器验证会话模式：该模式是在匿名会话模式提供的安全功能基础上，增加了服务器身份鉴别功能，但是对客户端不需要身份鉴别；

c) 双方验证会话模式：该模式是在服务器验证会话模式提供的安全功能基础上，增加了客户端身份鉴别功能；

d) 重用会话模式：a)、b)、c)三种会话模式用于从初始状态开始会话的情况，重用会话模式用于使用以前协商好的会话安全参数实例，重新建立一个会话的情况。

A.3.3 匿名会话模式

在匿名会话模式下，SSL 客户端和服务器间交互的消息和顺序如图 A.1 所示。

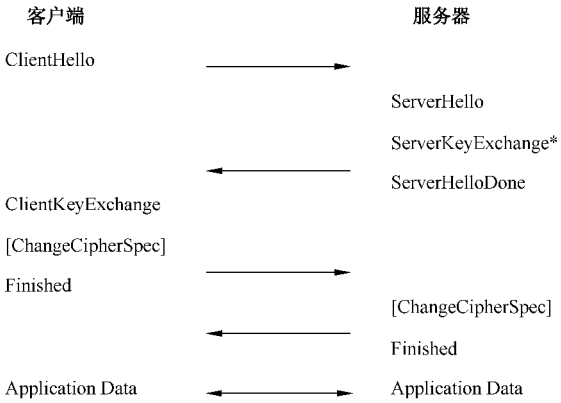


图 A.1 匿名会话模式消息流

客户端和服务器通过 ClientHello 消息和 ServerHello 消息协商好会话使用的密码算法组和数据压缩算法，密码算法组中包括密钥交换算法、数据加密算法和 MAC 算法。

根据选择的密钥交换算法，ServerKeyExchange * 消息（* 表示该消息在有的情况下不会出现）和 ClientKeyExchange 消息完成客户端和服务器之间的共享密钥协商。根据算法的不同，可能不需要服务器发送 ServerKeyExchange * 消息，而由 ClientKeyExchange 消息单独完成密钥协商的任务。

客户端和服务器通过 ChangeCipherSpec 消息指明此后的数据已经得到安全保护。

客户端和服务器通过 Finished 消息通知对方，有关协商已经完成，可以进行应用数据保护了，Finished 消息是应用当前安全服务保护的第一条消息。

根据 SSL 协议匿名会话模式消息流和出错处理情况，得到匿名会话模式下 SSL 客户端的行为状态如图 A.2 所示，匿名会话模式下 SSL 服务器的行为状态如图 A.3 所示。

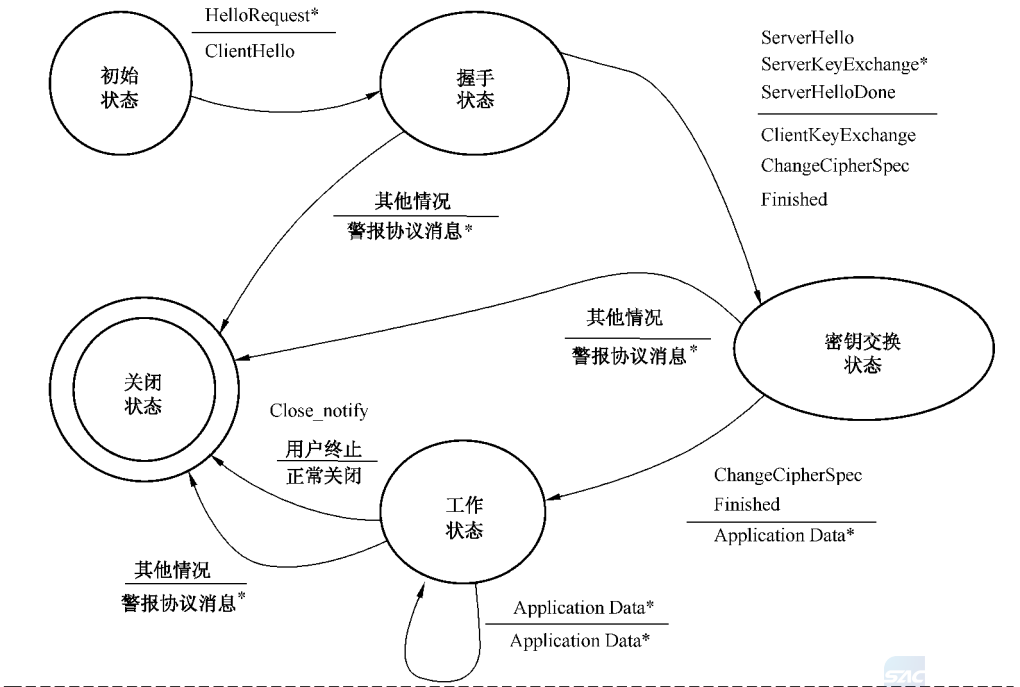


图 A.2 匿名会话模式下 SSL 客户端行为状态图

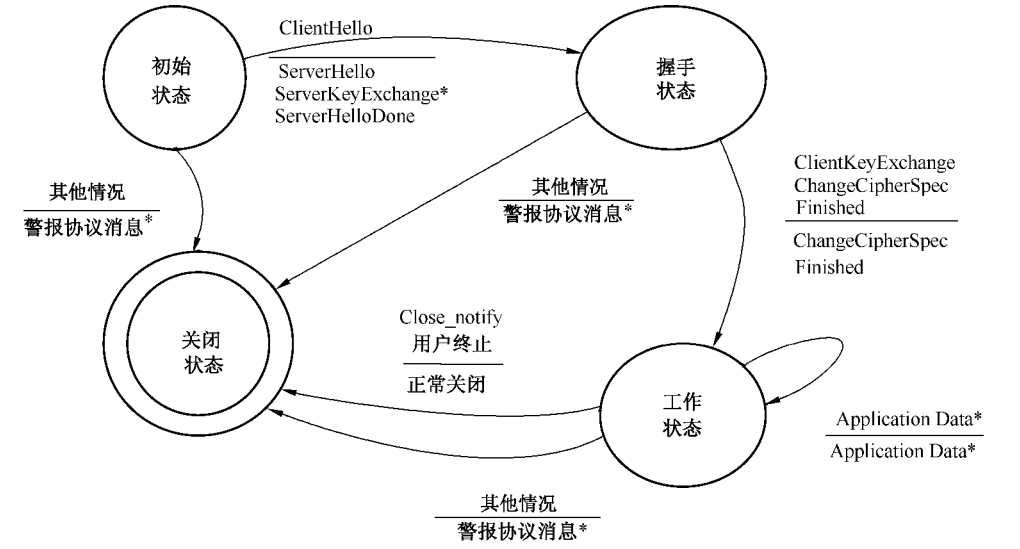


图 A.3 匿名会话模式下 SSL 服务器行为状态图

在匿名会话模式下的各种消息具体格式，见附录 A.4 SSL 消息格式规范。

A.3.4 服务器验证会话模式

在服务器验证会话模式下，SSL 客户端和服务器间交互的消息和顺序如图 A.4 所示。

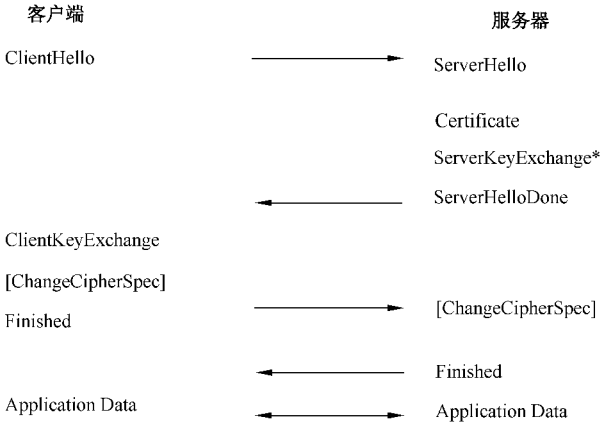


图 A.4 服务器验证会话模式消息流

服务器验证会话模式比匿名会话模式多了一条 Certificate 消息，服务器通过该消息向对方递交身份鉴别证书，通过证书验证实现服务器身份鉴别功能。身份证书格式以及有关的验证过程，应符合国家关于数字证书有关的标准和规定。其他消息的基本格式和含义与匿名会话模式一样。

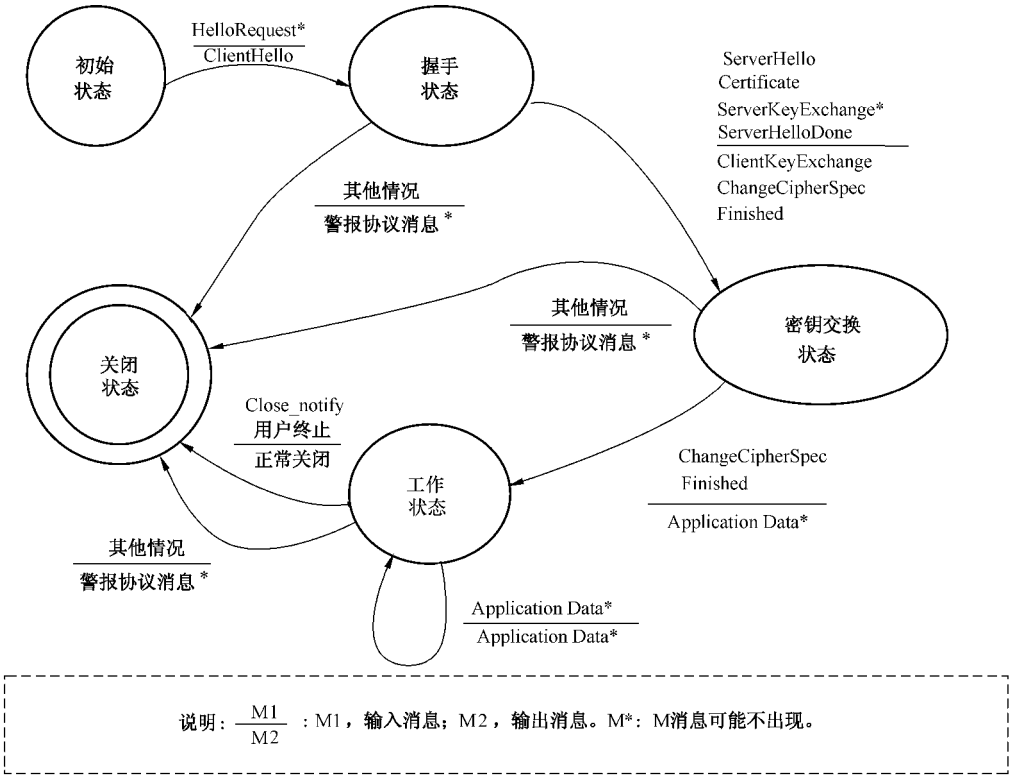


图 A.5 服务器验证会话模式下 SSL 客户端行为状态图

根据 SSL 协议服务器验证会话模式消息流和出错处理情况，得到服务器验证模式下 SSL 客户端的行为状态如图 A.5 所示，服务器验证会话模式下 SSL 服务器的行为状态如图 A.6 所示。

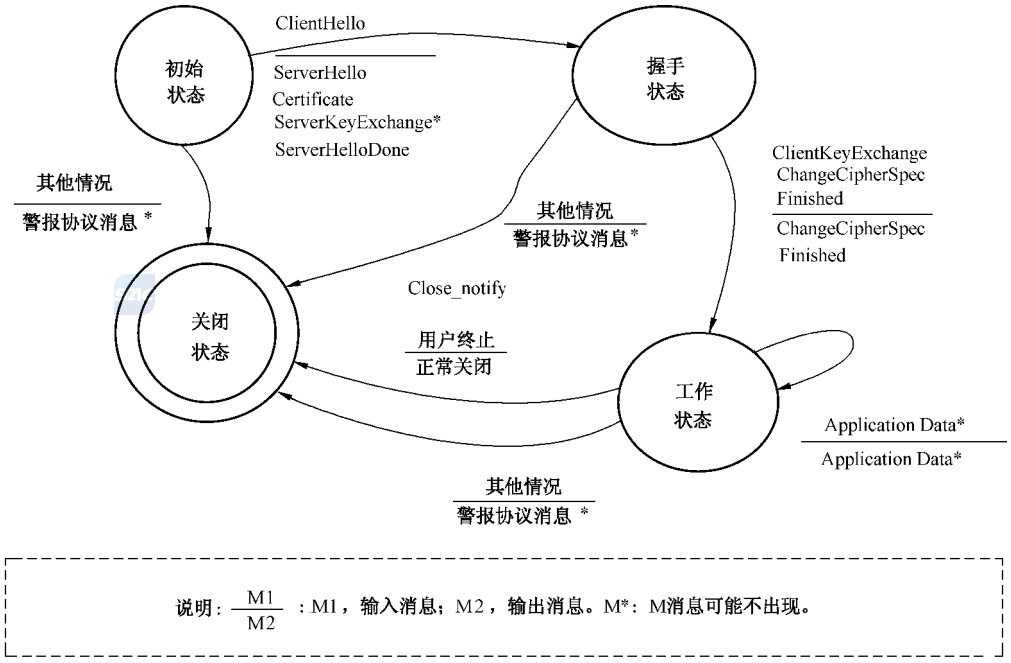


图 A.6　服务器验证会话模式下 SSL 服务器行为状态图

在服务器验证会话模式下各种消息的具体格式，见 A.4 SSL 消息格式规范。

A.3.5　双方验证会话模式

在双方验证会话模式下，SSL 客户端和服务器间交互的消息和顺序如图 A.7 所示。

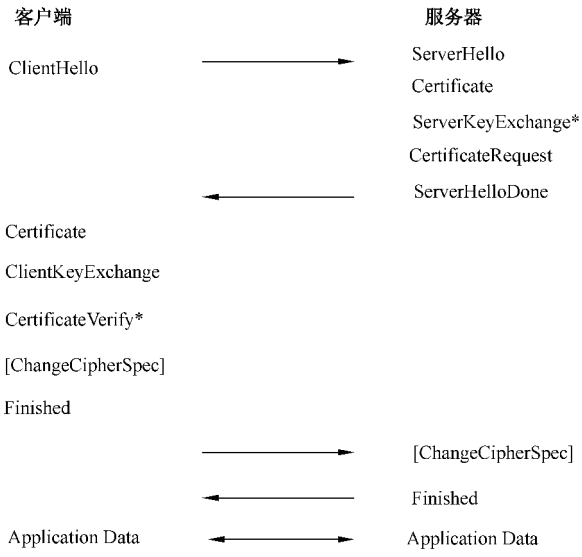


图 A.7　双方验证会话模式消息流

双方验证会话模式比服务器验证模式多了 CertificateRequest、客户端 Certificate 和 CertificateVerify * 三条消息，CertificateRequest 消息由服务器发出，请求验证客户端的身份，客户端 Certificate 消息向服务器递交身份鉴别证书，根据不同的身份证书类型，客户端有时需要发送证书验证 CertificateVerify * 消息以表明对证书的拥有权，而服务器的证书一般和密钥交换有关，所以不用发送服务器证书验证消息。身份证书格式以及有关的验证过程，应符合国家关于数字证书有关的标准和规定。其他消息的基本格式和含义与服务器验证模式相同。

根据 SSL 协议服务器验证会话模式消息流和出错处理情况，得到双方验证模式下 SSL 客户端的行为状态如图 A.8 所示，双方验证模式下 SSL 服务器的行为状态如图 A.9 所示。

双方验证会话模式下的各种消息具体格式，见附录 A.4 SSL 消息格式规范。

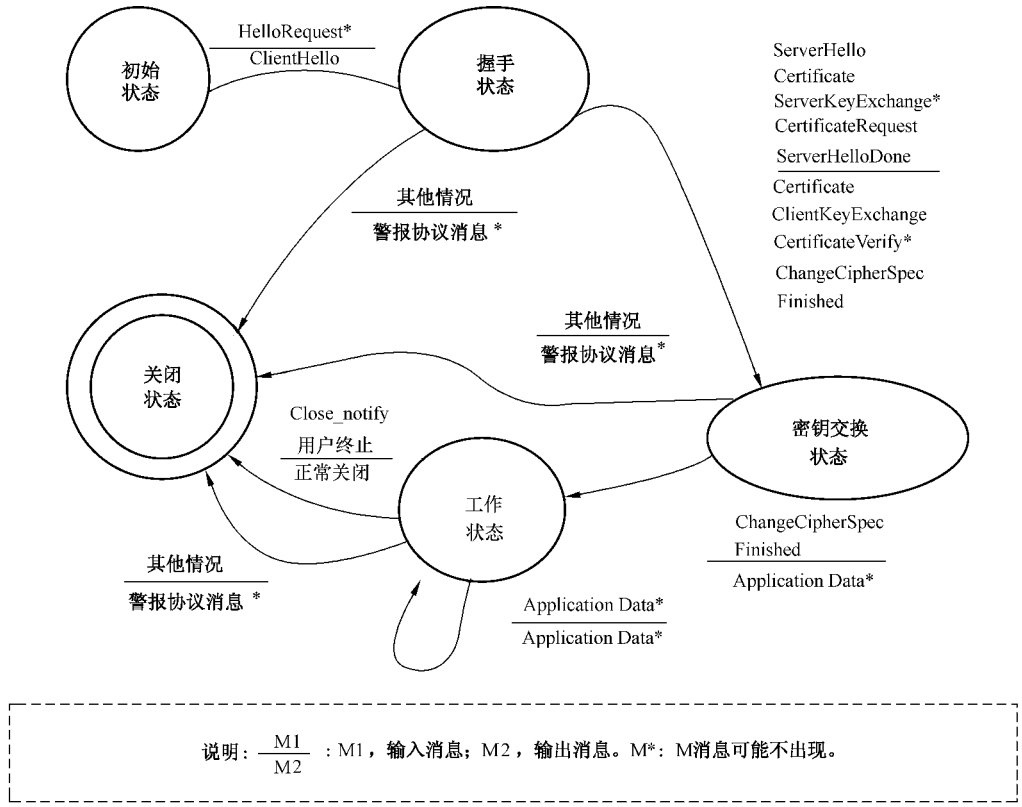


图 A.8　双方验证模式下 SSL 客户端行为状态图

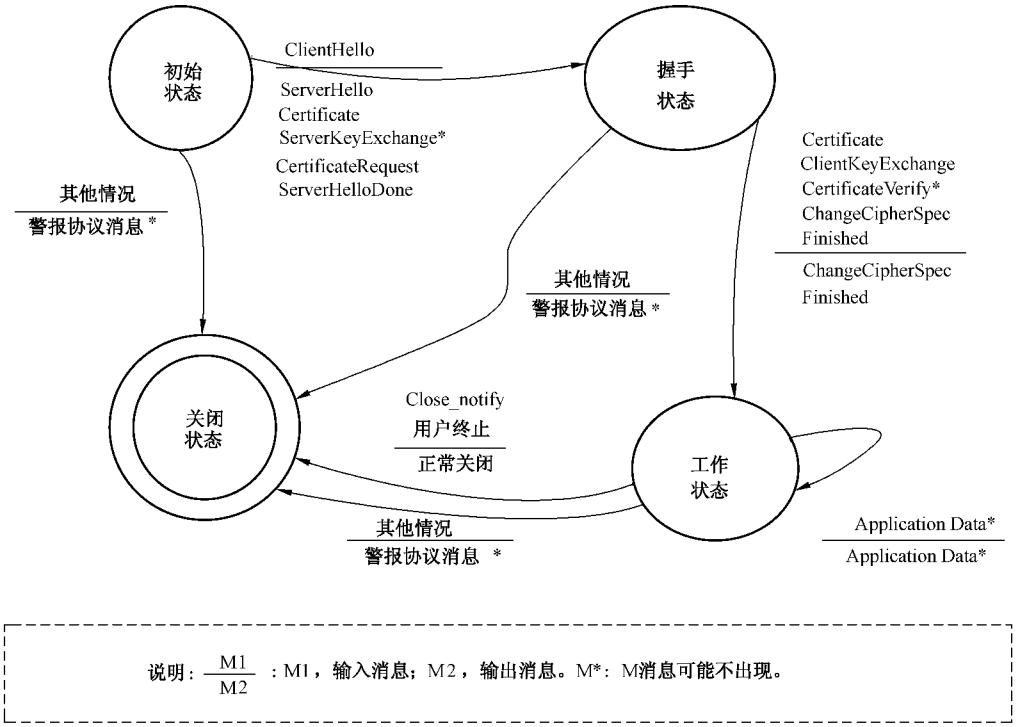


图 A.9　双方验证模式下 SSL 服务器行为状态图

A.3.6　重用会话模式

在重用会话模式下，SSL 客户端和服务器间交互的消息和顺序如图 A.10 所示。

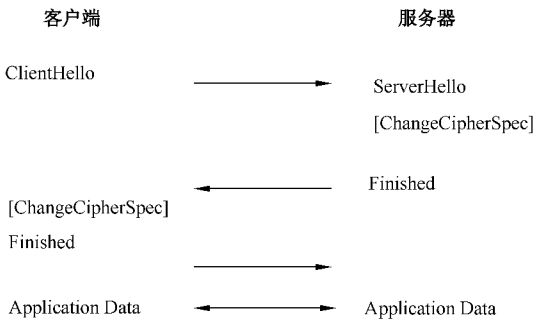


图 A.10　重用会话模式消息流

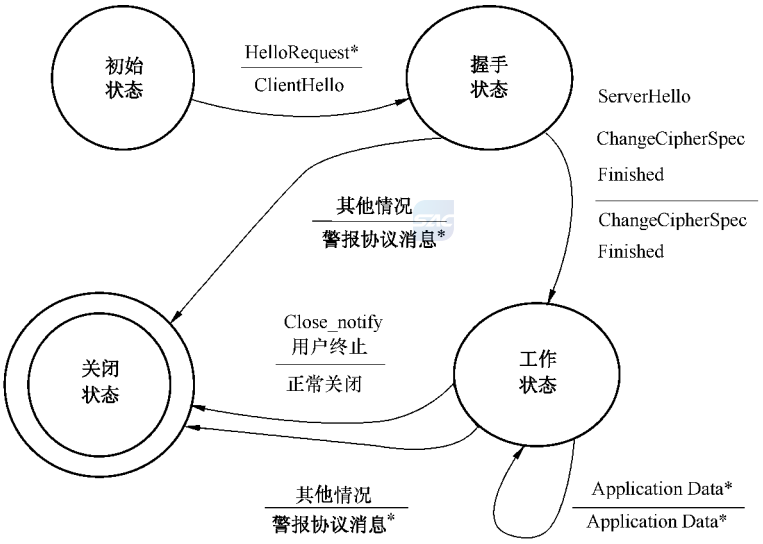
客户端通过在 ClientHello 消息中指定会话 ID，启动重用会话模式，该 ID 是以前通过 A.3.3、A.3.4、A.3.5 所描述的三种模式之一协商完成的 SSL 会话的标识。

一旦服务器同意启用原来协商好的 SSL 会话安全参数，服务器在 ServerHello 消息中传输被重用会话中的有关密码算法。

双方都同意使用原来 SSL 会话安全参数，则交换 ChangeCipherSpec 消息指明此后的数据已经得到安全保护。

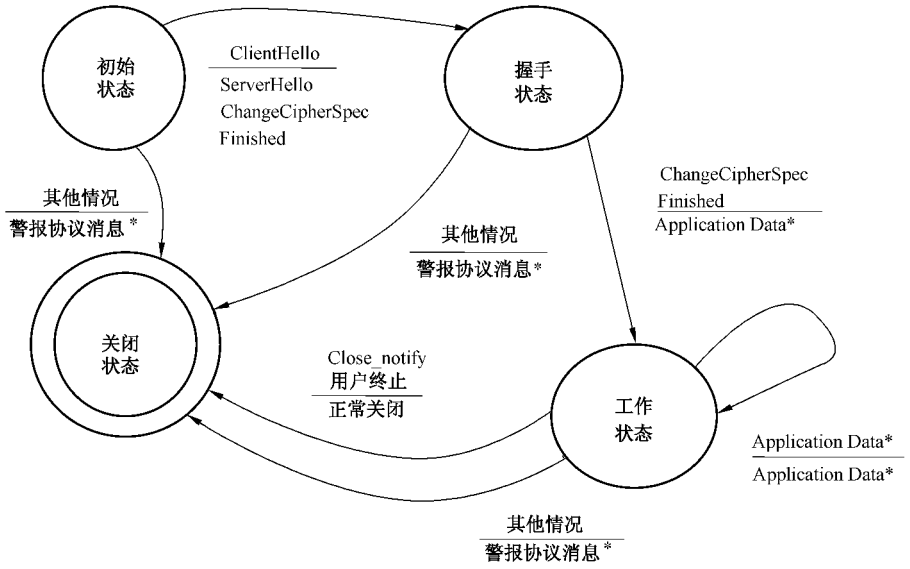
客户端和服务器通过 Finished 消息通知对方，有关协商已经完成，应用数据已经被保护了。

根据 SSL 协议会话重用模式消息流和出错处理情况，得到重用会话模式下 SSL 客户端的行为状态如图 A.11 所示，重用会话模式下 SSL 服务器的行为状态如图 A.12 所示。



说明：$\frac{M1}{M2}$ ：M1，输入消息；M2，输出消息。M*：M消息可能不出现。

图 A.11 重用会话模式下 SSL 客户端行为状态图



说明：$\frac{M1}{M2}$ ：M1，输入消息；M2，输出消息。M*：M消息可能不出现。

图 A.12 重用会话模式下 SSL 服务器行为状态图

重用会话模式下的各种消息具体格式，见 A.4 SSL 消息格式规范。

A.4 SSL 消息格式规范

A.4.1 SSL 消息基本格式

A.4.1.1 SSL 协议版本

SSL 中的消息都需要说明所使用协议版本号，格式如下。

```
struct {
   uint8 major, minor;
} ProtocolVersion;
```

SSL 协议中包括 3 种类型的版本号，不同的版本号对应的值如下：

a) SSL3.0：major＝3，minor＝0；

b) SSL3.1/TLS1.0：major＝3，minor＝1；

c) TLS1.1：major＝3，minor＝2。

A.4.1.2 SSL 消息类型

SSL 中的消息包括 4 种类型：

a) change_cipher_spec(更改密码组规范消息)：该类消息通知通信对方，此后的所有消息都使用协商的好的安全参数实例对会话进行保护；

b) alert(警报协议消息)：该类消息用于通知一些消息或者报告一些错误；

c) handshake(握手协议消息)：该类型消息用于协商会话的安全参数实例；

d) application_data(应用数据消息)：该类型消息用于对使用 SSL 进行数据传输的应用层数据进行安全保护。

消息类型说明格式如下。

```
enum {
     change_cipher_spec(20), alert(21), handshake(22),
     application_data(23), (255)
} ContentType;
```

A.4.1.3 SSL 消息格式

SSL 消息格式如下所示。

```
struct {
    ContentType type;
    ProtocolVersion version;
    uint16 length;
    opaque message_data[Massage.length];
} Message;
```

该消息格式中各字段的含义如下：

a) type：消息类型，其值为 A.4.1.2 中说明的四种类型之一；

b) version：SSL 协议版本，其值为 A.4.1.1 中说明的三种类型之一；

c) length：消息数据长度，如果消息没有加密，则长度不能够超过 2^{14}，如果消息加密，则长度不能够超过 $2\ 048+2^{14}$；

d) message_data：对应消息类型的数据。

A.4.2 握手协议消息

A.4.2.1 握手协议消息类型

如果 SSL 消息中的消息类型是 handshake(22),则 SSL 消息中的数据 message_data 就是握手协议消息类型,一个 SSL 消息中可能包含了多个握手协议消息。

握手协议消息所包含的类型如下所示。

```
enum {
    hello_request(0),
    client_hello(1),
    server_hello(2),
    certificate(11),
    server_key_exchange(12),
    certificate_request(13),
    server_hello_done(14),
    certificate_verify(15),
    client_key_exchange(16),
    finished(20),
    (255)
} HandshakeType;
```

每一个握手消息的基本格式如下所示。

```
struct  {
    HandshakeType msg_type;      /* 握手消息类型 */
    uint24 length;               /* 消息长度 */
    handshake_body[Handshke.Length];   /* 消息体 */
} Handshake;
```

A.4.2.2 Hello 消息

A.4.2.2.1 Hello 消息类型

SSL 会话双方(客户端和服务器)通过 Hello 消息开始协商一个新的会话,通过 Hello 消息的交互,会话双方在密钥交换算法、数据加密算法、MAC 算法和数据压缩算法等方面达成一致。

Hello 消息包括三种类型:

a) HelloRequest:服务器通过该消息告诉客户端希望重新进行会话协商;

b) ClientHello:客户端通过该消息,声明其支持的密钥交换算法、数据加密算法、MAC 算法和数据压缩算法;

c) ServerHello:服务器通过该消息,告诉客户端双方协商好的密钥交换算法、数据加密算法、MAC 算法和数据压缩算法。

A.4.2.2.2 HelloRequest

HelloRequest 由服务器发向客户端,意味着服务器希望客户端重新进行会话协商。HelloRequest 消息体格式如下:

```
struct { } HelloRequest;
```

服务器可以在任何时候发送该消息,客户端如何正在协商会话,则忽略该消息;在其他情况下,不同

SSL 协议版本处理方式可以不一样，处理情况如下：

a) SSL3.0：客户端发送 ClientHello 消息开始一个新的会话，或者忽略该消息。

b) TLS1.0 和 TLS1.1：客户端发送一个 ClientHello 消息开始一个新的会话，或者忽略该消息，或者发送一个 no_renegotiation 的警报协议消息。

A.4.2.2.3 ClientHello

客户端可以在初始会话时，向服务器发送该消息，也可以作为服务器 HelloRequest 消息的响应消息，该消息开始一个 SSL 会话协商。

客户端通过 ClientHello 消息声明其支持的密钥交换算法、数据加密算法、MAC 算法和数据压缩算法。

密钥交换算法、数据加密算法和 MAC 算法的一个组合称为一个密码组(CipherSuite)，对应每一个支持的密码都由一个 16 位的整数来标识。SSL 实现时所选择的密码组，必须符合国家的有关标准和规定。

ClientHello 消息可声明客户端支持的多个密码组，按照优先级排列，密码组结构如下所示。

```
uint8 CipherSuite[2];
```

ClientHello 消息格式如下所示。

```
struct {
    ProtocolVersion client_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suites<2..2^16-1>;
    CompressionMethod compression_methods<1..2^8-1>;
} ClientHello;
```

各字段的基本含义如下：

a) client_version：客户端 SSL 协议版本号，具体值含义见 A.4.1.1。

b) random：客户端产生的随机数，结构如下：

```
struct {
    uint32 gmt_unix_time;
    opaque random_bytes[28];
} Random;
```

gmt_unix_time 是当前时间，用 32 比特位的标准 UNIX GMT 时间(从 1970 年 1 月 1 号零点以来，时间所经过的秒数)；random_bytes 是一个 28 字节的随机数。

c) session_id：如果该值不为 0 字节，客户端启动了一个重用会话模式，session_id 表示希望重用的会话的标识符；如果该值是 0 个字节，则表示开始一个新的会话协商。

d) cipher_suites：客户端所支持的密码组，按照优先级顺序排列，服务器从中选择一个服务器也支持的密码组，该密码组就是协商好的密码组；如果服务器无法从中选择一个密码组，则返回一个握手失败的警报协议消息。

e) compression_methods：客户端所支持的压缩算法。

A.4.2.2.4 ServerHello

服务器接收到客户端发送来的 ClientHello 消息时，如果能够从 ClientHello 中声明的密码组中选择出一组服务器也支持的密码组，则发送一个 ServerHello 消息，否则发送一个握手失败的警报协议消息。

ServerHello 消息格式如下。

```
struct {
    ProtocolVersion server_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suite;
    CompressionMethod compression_method;
} ServerHello;
```

各字段的含义如下：

a) server_version：服务器 SSL 协议版本号，具体值含义见 A.4.1.1。

b) random：服务器产生的随机数，具体结构见 A.4.2.2.3 。

c) session_id：本次会话的标识，小于或者等于 32 字节长度的字符串。

d) cipher_suite：服务器和客户端之间协商好的密码组。

e) compression_method：服务器和客户端之间协商好的压缩算法。

A.4.2.3 服务器 Certificate

在服务器验证会话模式或者双方验证会话模式下，服务器需要发送 Certificate 消息，该消息紧跟 ServerHello 消息后面。

服务器发送 Certificate 消息传输一个身份证书，通过发送服务器身份证书，主要有两个目标，一是对服务器身份进行鉴别，二是传输一个符合密钥交换算法的密钥(一般来说，该密钥是服务器的公开密钥)。

SSL 协议应用采用的身份证书机制应符合国家有关标准和规定。

A.4.2.4 ServerKeyExchange

如果密钥交换算法中需要服务器发送密钥交换材料，则服务器发送该消息。该消息紧跟 Certificate 消息后面(如果没有服务器 Certificate 消息，则紧跟 ServerHello 消息后面)。

ServerKeyExchange 消息中包含的具体密钥交换材料，要和具体的密钥交换算法有关，SSL 实现的密钥交换算法应符合国家的有关标准和规定。

A.4.2.5 CertificateRequest

在双方验证会话模式下，服务器要发送 CertificateRequest 消息，请求客户端的身份证书。

CertificateRequest 消息紧跟 ServerKeyExchange 消息后，如果没有 Server KeyExchange 消息，则紧跟服务器 Certificate 消息后。

CertificateRequest 消息的格式如下：

```
struct {
    ClientCertificateType certificate_types<1..2^8-1>;
    DistinguishedName certificate_authorities<3..2^16-1>;
} CertificateRequest;
```

各字段的基本含义如下：

a) certificate_types：请求的证书类型列表，列表中的每一项是一个字节的整数，代表一种证书类型，SSL 协议应用应根据国家关于数字证书的标准和规定进行定义；

b) certificate_authorities：服务器接受的 CA 列表。

A.4.2.6 ServerHelloDone

除了重用会话模式外，一个 SSL 会话过程中，服务器都需要发送 ServerHelloDone 消息。

ServerHelloDone 消息表明服务器已完成了会话协商的准备工作，等待客户端的会话协商完成响应。ServerHelloDone 消息是服务器响应 ClientHello 消息的最后一个消息。

ServerHelloDone 消息格式如下：

```
struct { } ServerHelloDone;
```

A.4.2.7 客户端 Certificate

如果服务器发送了 CertificateRequest，请求验证客户端的身份，则客户端需要发送客户端 Certificate 消息。

如果需要发送该消息，则客户端在收到 ServerHelloDone 消息后，第一个发送该消息。

该消息格式和服务器 Certificate(见 A.4.2.3)消息一致。

A.4.2.8 ClientKeyExchange

在任何情况下，客户端都需要发送该消息，以完成客户端和服务器之间的密钥协商。

ClientKeyExchange 消息紧跟客户端 Certificate 消息后面(如果没有客户端 Certificate 消息，则客户端接收到 ServerHelloDone 消息后，发送的第一个消息)。

ClientKeyExchange 消息中包含的具体密钥交换材料，要和具体的密钥交换算法有关，SSL 协议应用采用的密钥交换算法应符合国家的有关标准和规定。

A.4.2.9 CertificateVerify *

如果需要对客户端证书的有效性进行验证，客户端需要发送该消息，该消息紧跟 ClientKeyExchange 消息后面。

该消息的具体内容和证书要求有关，SSL 协议应用应根据国家关于数字证书的标准和规定进行定义。

A.4.2.10 Finished

一旦更改密码组规范消息(见 A.4.3)已经发出，该消息紧跟其后，以验证密码协商是否成功。该消息是第一个被协商好的密码算法保护的消息。

Finished 消息的具体内容和选择的 MAC 算法有关。

A.4.3 更改密码组规范消息

该消息用来通知对方，此后的消息，都是使用新协商的密码组和密钥来保护。

该消息的基本格式如下：

```
struct {
    enum { change_cipher_spec(1), (255) } type;
} ChangeCipherSpec;
```

A.4.4 警报协议消息

A.4.4.1 警报协议消息一般格式

警报协议消息用来通知通信过程中的一些情况。

警报协议消息的基本格式如下：

```
struct {
    AlertLevel level;
    AlertDescription description;
} Alert;
enum { warning(1), fatal(2), (255)} AlertLevel;
```

对于 AlertDescription 的描述，SSL3.0 以后的版本（TLS1.0 和 TLS1.1）有所区别，详细区别分别见 A.4.4.2 SSL3.0 警报类型和 A.4.4.3 TLS1.0/TLS1.1 警报类型。

A.4.4.2 SSL3.0 警报类型

SSL3.0 警报类型如下：

```
enum {
      close_notify(0),
      unexpected_message(10),
      bad_record_mac(20),
      decompression_failure(30),
      handshake_failure(40),
      no_certificate(41),
      bad_certificate(42),
      unsupported_certificate(43),
      certificate_revoked(44),
      certificate_expired(45),
      certificate_unknown(46),
      illegal_parameter (47),
      (255)
} AlertDescription;
```

这些消息的基本含义如下：

a) close_notify：该警报类型用于通知对方关闭当前会话，在正常关闭会话的过程中，客户端和服务器必须发送该消息。

b) unexpected_message：接收到非期望的消息时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

c) bad_record_mac：接收到消息 MAC 验证失败时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

d) decompression_failure：数据解压缩失败时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

e) handshake_failure：无法完成握手协议时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

f) no_certificate：如果对方请求的证书，本地无法满足，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

g) bad_certificate：如果收到一个破坏的证书，也就是说，证书的签名验证失败的话，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

h) unsupported_certificate：如果收到的证书格式，本地不支持，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

i) certificate_revoked：如果收到的证书已经被其颁发者撤消，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

j) certificate_expired：如果收到的证书已经过期，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

k) certificate_unknown：在处理证书过程中碰到的其他出错情况，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

l) illegal_parameter：如果握手协议数据在一特定域的值超过了规范规定的值，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

A.4.4.3 TLS1.0/TLS1.1 警报类型

TLS1.0/TLS1.1 在 SSL3.0 的基础上，对一些警报类型进行的调整。

删除的警报类型有：

a) no_certificate(41)

新增的警报类型有：

a) decryption_failed：解密出来的数据不符合密码算法的要求(数据长度不是某数的整数倍，或者填充数据的值不符合要求)，一旦出现这样的情况，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

b) record_overflow：接收到的数据长度，超过了规定的长度，一旦出现这样的情况，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

c) unknown_ca：当找不到信任 CA 的证书时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

d) access_denied：用户递交了有效的证书，但是根据访问控制策略，协商不能够继续时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

e) decode_error：接收到消息，解码失败，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

f) decrypt_error：解密操作失败，包括验证签名过程、数据解密过程中出现错误，则发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

g) protocol_version：协议版本类型不能够识别，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

h) insufficient_security：当服务器要求的安全强度更高的密码组时，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

i) internal_error：一方在处理数据时，出现内部错误时(如内存分配失败)，发送该警报消息，该消息是严重的警报，一旦出现该类警报，则会话终止。

j) user_canceled：协商好会话后，用户取消该会话，则发送该消息，该消息一般紧跟 close_notify 消息后面，该消息是警告级。

k) no_renegotiation：当客户端收到 HelloRequest 消息时，此时又不需要进行重新协商时，发送该警报消息；当服务器接收到客户端重新协商的 ClientHello 时，此时又不需要进行重新协商时，发送该警报消息。

参 考 文 献

[1] Draft302 SSL 协议 3.0 版(The SSL Protocol Version 3.0)

[2] RFC2246 TLS 协议 1.0 版本(SSL 协议 3.1 版)(The TLS Protocol Version 1.0)

[3] RFC3546 传输层安全(TLS)扩展(Transport Layer Security (TLS) Extensions)

[4] RFC4346 传输层安全(TLS)协议 1.1 版 (The Transport Layer Security (TLS) Protocol Version1.1)

[5] SSL VPN 技术规范,国家密码管理局发布,2009 年 1 月

GB/T 28457—2012

中 华 人 民 共 和 国
国 家 标 准
SSL 协议应用测试规范
GB/T 28457—2012
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲 2 号(100013)
北京市西城区三里河北街 16 号(100045)
网址:www.gb168.cn
服务热线:010-68522006
2012 年 10 月第一版
*
书号:155066·1-45612

GB/T 28457-2012