- 第 3 部分通过创建本地套接字、设置远端地址信息并连接到远端服务器，建立了一条到服务器端口 443 的 TCP 连接。
- 一旦 TCP 连接建立起来，就用 SSL_new 和 SSL_set_fd 将 SSL 层附加到 TCP 连接之上，并调用 SSL_connect 与服务器进行 SSL 握手。第 4 部分完成时，我们就建立了一个已选好密码且交换过证书的可运行的 SSL 信道。
- 第 5 部分打印了选中的批量加密密码值。
- 第 6 部分打印了服务器回送的 X.509 证书中包含的部分信息，其中包括与证书持有者和颁发证书的组织有关的信息。OpenSSL 库没有对服务器证书中的信息作任何特殊的处理。实际的 SSL 应用程序，比如 Web 浏览器会对证书进行一些完整性检查，以确保证书是正确签发的，且是来自正确主机的。我们在 14.7.6 节讨论了浏览器对服务器证书所做的处理。
- 此时，我们的 SSL 连接就已经可以用于安全数据的传输了。在第 7 部分中，用 SSL_write 在 SSL 信道上发送了简单的 HTTP 请求 GET / HTTP/1.0，然后关闭了连接的输出端。
- 在第 8 部分中，用 SSL_read 从连接上读回响应，并将其打印到屏幕上。SSL 层负责所有的加密和解密工作，因此可以直接读写普通的 HTTP 命令。
- 最后，在第 9 部分进行了一些清理工作。

更多与 OpenSSL 库有关的信息请参见 http://www.openssl.org。

14.8.3 执行OpenSSL客户端

下面显示了指向安全服务器时这个简单 HTTP 客户端的输出。在这个例子中，客户端指向了摩根士丹利的在线证券主页。在线交易公司都在广泛使用 HTTPS。

333

```
% https_client clients1.online.msdw.com
(1) SSL context initialized

(2) 'clients1.online.msdw.com' has IP address'63.151.15.11'

(3) TCP connection open to host 'clients1.online.msdw.com', port 443

(4) SSL endpoint created & handshake completed

(5) SSL connected with cipher: DES-CBC3-MD5

(6) server's certificate was received:

      subject: /C=US/ST=Utah/L=Salt Lake City/O=Morgan Stanley/OU=Online/CN=
        clients1.online.msdw.com
      issuer: /C=US/O=RSA Data Security, Inc./OU=Secure Server Certification
        Authority
(7) sent HTTP request over encrypted channel:
```