**实验二 基于OpenSSL的安全Web服务器程序**

1. **实验目的**

（1）理解HTTPS协议与SSL协议的基本工作原理；

（2）掌握使用OpenSSL编程的方法；

（3）掌握安全Web系统设计的基本设计与编程方法。

1. **实验要求**

（1）在Linux平台上利用OpenSSL库，编写一个Web Server程序；

（2）Server程序要能够并发处理多个请求，要求至少能够支持HTTPS协议下最基本的Get命令，可以增强Web Server的功能，如支持Head、Post以及Delete命令等；

（3）编写必要的客户端测试程序，用于发送HTTPS请求并显示返回结果。

1. **实验内容**

根据本章所学的知识，在Linux平台上利用OpenSSL实现安全的Web Server。Server能够并发处理多个请求，要求至少能支持Get请求。可以增强Web Server的功能，如支持Head、Post以及Delete命令等。

编写必要的客户端测试程序，用于发送HTTPS请求并显示返回结果，也可以使用一般的Web浏览器测试。

1. **实验原理**

实验程序可分为两个部分：初始化模块，即通过编译及函数调用对OpenSSL库进行初始化并且创建上下文环境；Web服务模块，即基于SSL机制利用OpenSSL库函数实现HTTPS的服务。程序整体流程如下图：



**（1）初始化模块**

在开启HTTPS服务之前，服务器端只需要初始化OpenSSL库和创建上下文环境。在此过程中，会用到如下函数：

SSL\_library\_init(); //加载OpenSSL将会用到的算法

SSL\_load\_error\_strings(); //加载错误字符串

SSL\_METHOD \*meth();

SSL\_CTX \*ctx; //SSL\_CTX对象

meth=SSLv23\_method(); //相应的SSL结构使用的是SL2.0、3.0但可以回到SSL2.0

ctx=SSL\_CTX\_new(mesh); //创建一个上下文环境

SSL\_CTX\_use\_certificate\_chain\_file(ctx, ”server.pem”); //指定所使用的证书文件

SSL\_CTX\_set\_default\_passwd\_cb(ctx, password\_cb); //设置密码回调函数

SSL\_CTX\_use\_PrivateKey\_file(ctx, “server.pem”, SSL\_FILETYPE\_PEM); //加载私钥文件

SSL\_CTX\_load\_verify\_locations(ctx, “root.pem”, 0); //加载受信任的CA证书

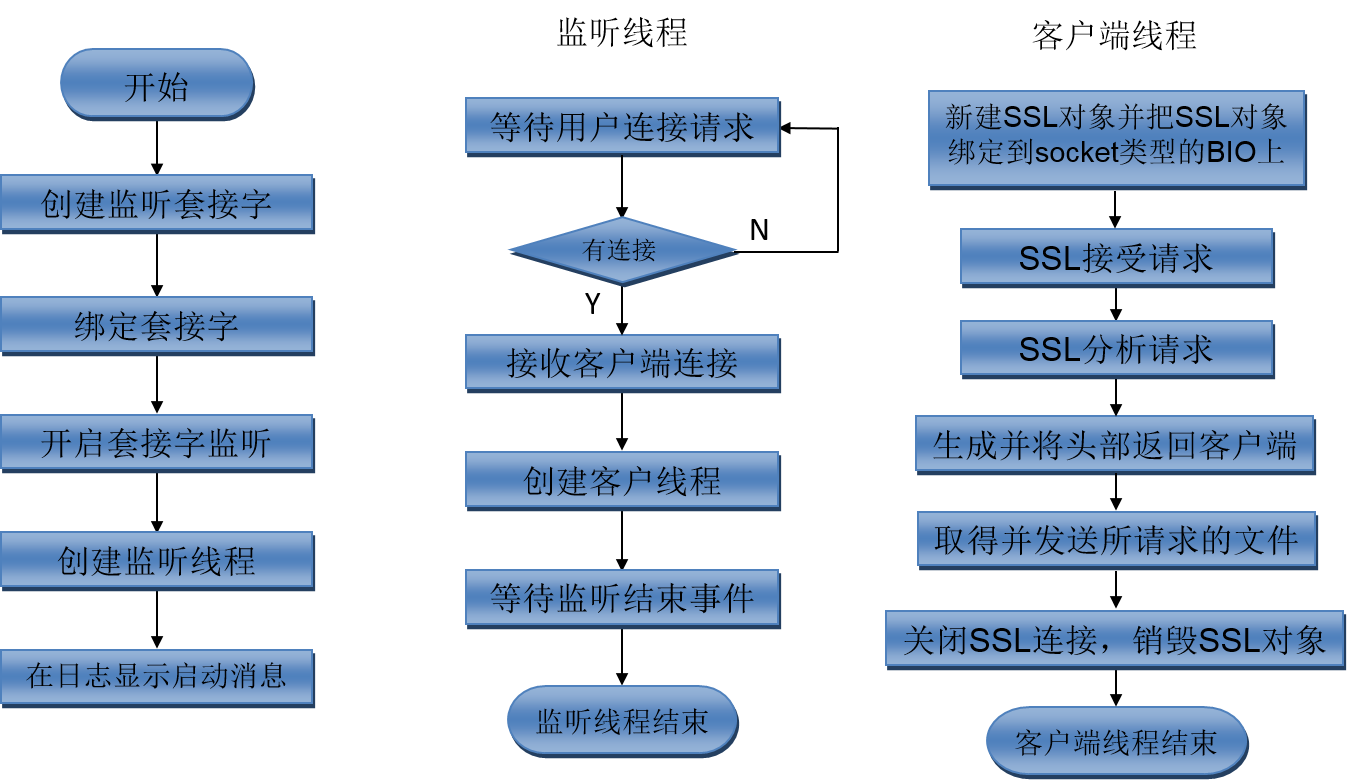
Load\_dh\_params(ctx, “dh1024.pem”);

//当使用RSA算法鉴别的时候，会有一个临时的DH密钥磋商发生。这样会话数据将用

//这个临时的秘钥加密，而证书中的密钥作为签名。

**（2）Web服务模块**

Web服务模块是本程序的核心部分，其执行流程如下图所示：



Web服务模块在类CHttpProtocol中实现。该类中封装了HTTPS中与本程序相关的操作，主要包括监听线程函数以及客户端函数中需要调用的子函数的定义及实现。其中，比较核心的子函数包括SSL分析请求，将响应头部返回给客户端以及将文件发送回客户端等。

1. **实验环境**

（1）Linux系统：Ubuntu 14.04 LTS

（2）Openssl版本：OpenSSL 1.0.1f

（3）g++ : 4.8.4

（4）浏览器: firefox

1. **实验步骤**

**步骤1** 配置OpenSSL库，执行sudo apt-get install libssl-dev安装。

**步骤2** 制作自己的WebServer目录，目录下应有index.html文件（自行编写）以及favicon.ico文件。例如，制作/home/WebServer目录（也可以自定义路径），将index.html文件以及favicon.ico文件放入该文件夹。

**步骤3** 在HttpProtocol.cpp文件中第9行处，将m\_strRootDir设置成自己的WebServer目录路径。例如，m\_strRootDir="/home/WebServer";

**步骤4** 在命令行模式下，切换到代码所在目录，输入make后回车，完成代码的编译。若未安装g++，使用sudo apt-get install g++进行安装。

**步骤5** 在命令行中输入./MyWebServer 运行程序。

**步骤6** 打开浏览器，在地址栏输入https://[服务器IP地址]:8000，即可登录服务器页面。例如，<https://localhost:8000>。**注意：**如果浏览器提示此链接不受信任，需要添加例外后再进行访问。添加例外后，可以在firefox中查看添加的认证，位置为preferences -> privacy&security -> view certificates -> servers。

1. **预期实验结果**

（1）能够访问服务器页面。

（2）控制台中有如下记录：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Server starts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Starting ListenThread...

Starting ClientThread...

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Starting ClientThread...

IF...

Request received!!

GET / HTTP/1.1

Host: 127.0.0.1:8000

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:37.0) Gecko/20100101 Firefox/37.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Connection: keep-alive

Cache-Control: max-age=0

/home/WebServer/index.html

SSLSendHeader successfully!

Sending..............................

complete

File sent!!Closing socket!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*