基于 HTTPS 的数据加密

本小节为可选章节，因为 HTTPS 证书需要域名（域名需另行购买）才能申请。有域名的读者可以按步骤实践，没有域名的读者，只需要了解即可。

HTTPS 与 HTTP 区别

在前面的小节中，客户端与服务器端的请求响应都是用的 HTTP，HTTP 和 HTTPS 有什么区别呢?

HTTP 协议传输的数据都是未加密的，也就是明文的，因此使用 HTTP 协议传输隐私信息非常不安全，为了保证这些隐私数据能加密传输，网景公司设计了 SSL（Secure Sockets Layer）协议用于对 HTTP 协议传输的数据进行加密，从而就诞生了 HTTPS。简单来说，HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 HTTP 协议安全。

HTTPS 和 HTTP 的区别主要如下：

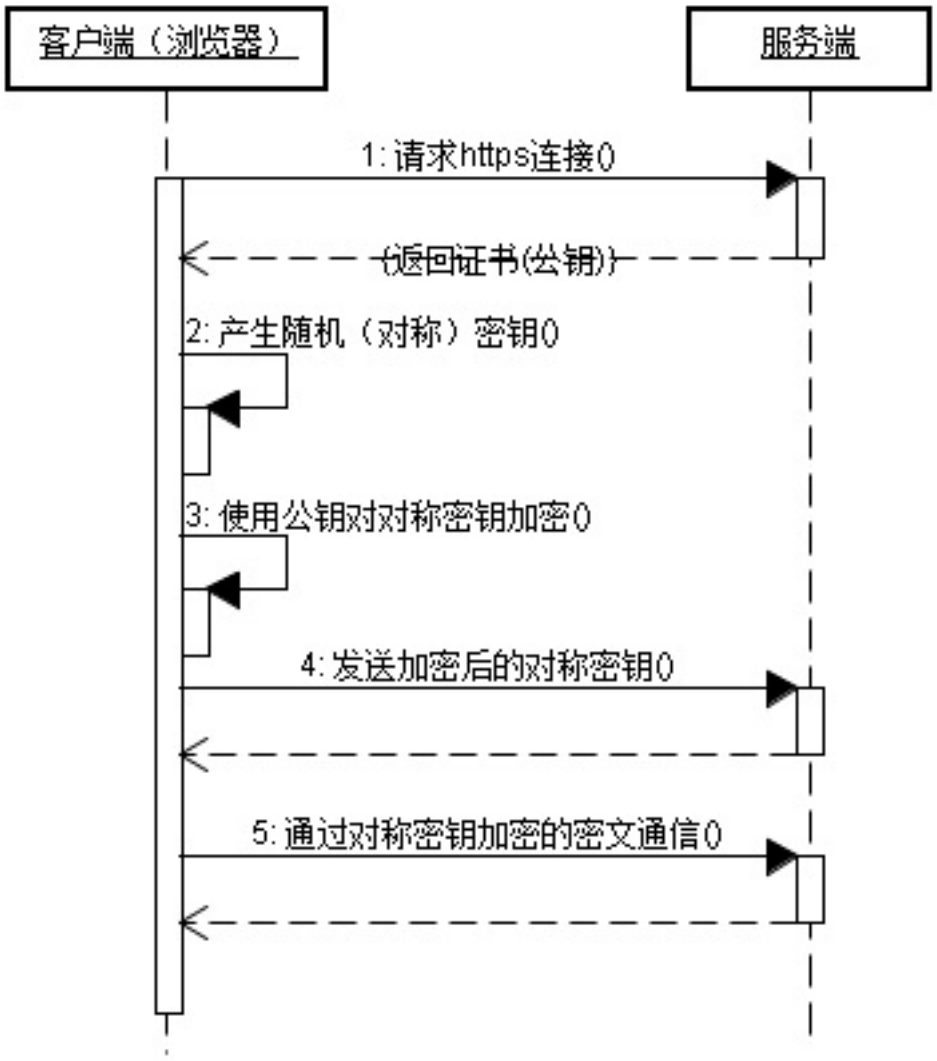
1. HTTPS 协议需要到 CA 中心申请证书
2. HTTP 是超文本传输协议，信息是明文传输，HTTPS 则是具有安全性的 SSL 加密传输协议
3. HTTP 和 HTTPS 使用的是完全不同的连接方式，前者默认是 80，后者是 443
4. HTTP 的连接很简单，是无状态的；HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，比 HTTP 协议安全

HTTP 由于是明文传输未加密，缺点可见一斑。这里插入一个小故事，在笔者开发第一款 App 的时候，为了提高效率，采用了 HTTP，在 客户端和服务器调试期间，笔者发现客户端的最下面，经常会莫名其妙地出现垃圾广告，一开始并不清楚为啥会出现这种情况，客户端和服务器端还花好长一段时间进行调试定位。最后发现是运营商的问题，广告也是运营商嵌入的，这就是不加密的后果：HTTP 被劫持了。后面改为 HTTPS，整个世界也就清静了。

HTTPS 工作原理

这里涉及很多新的概念，如公钥和私钥。简单的理解即为，我们的服务器端需要安装 CA 证书（证书下载后面会讲解），证书包含两个东西，一个是私钥，一个是公钥，私钥就是自己留着的，别人不会知道，公钥是别人使用 HTTPS 请求时，发给别人的密钥。当客户端需要发送加密报文时，会使用服务器端给的公钥进行加密，此时在网络中传输的就是一串无序的字符串。当报文被服务器端接收到时，服务器端使用私钥进行解密，这样就能保证整个链路的安全性。关于公钥和私钥，这里有一篇有趣的讲解供读者参考（公钥与私钥，HTTPS 详解 (https://www.cnblogs.com/shijingjing07/p/5965792.html)），本节作为简单的抛砖引玉，不作过多的阐述。下面看一下整个通信流程。

客户端在使用 HTTPS 方式与服务器端通信时有以下几个步骤，如图所示。



1. 客户使用 HTTPS 的 URL 访问服务器，要求与服务器建立 SSL 连接
2. 服务器收到客户端请求后，会将站点的证书信息（证书中包含公钥）传送一份给客户端
3. 客户端与服务器开始协商SSL连接的安全等级，也就是信息加密的等级
4. 客户端的浏览器根据双方同意的安全等级，建立会话密钥，然后利用服务器端的公钥将会话密钥加密，并传送给客户端
5. 服务器利用自己的私钥解密出会话密钥

下载证书

如上所述，首先需要申请下载证书，并将其存放在服务器端。目前安全性较高的数字证书都是付费的。读者可以根据自身项目的诉求，选择不同的证书级别，个人或者小微企业可选择使用免费的数字证书。由于我们只是 Demo，这里选择免费证书。免费的证书可以直接上公有云提供商下载，如腾讯云、阿里云等。本小册以腾讯云为例。

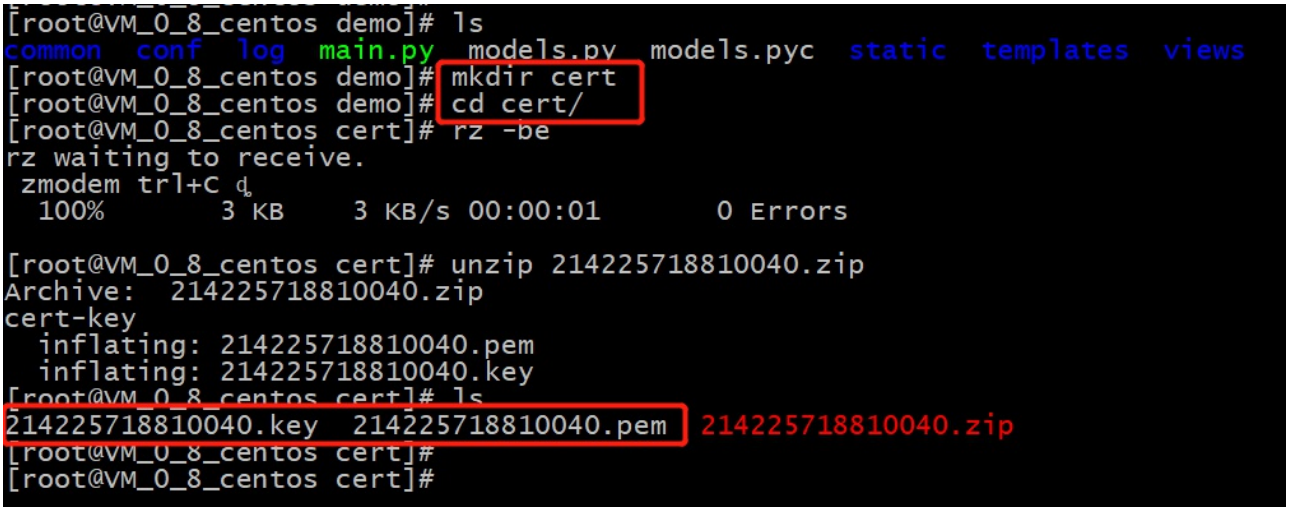
申请证书

登录腾讯云，输入如下链接，申请“域名型免费版(DV)”：
https://buy.cloud.tencent.com/ssl?fromSource=ssl

按照步骤一步步完成购买。

下载上传

将证书从腾讯云上下载下来（214225718810040.zip），并将其上传到服务器上。假定证书也放在 demo 目录下，在 demo 目录下创建 cert 目录。并将其解压至此目录。



配置 Nginx

正如前面介绍 HTTPS 时所述，HTTPS 使用的是 443 端口，此时需要修改 Nginx 监听的端口为 443。另外，需要在 Nginx 的配置文件中指定 HTTPS 证书的路径。配置 Nginx 的 server 如下：

```
listen 443;
server_name  _;
ssl on;
root html;
index index.html index.htm;
ssl_certificate   cert/214225718810040.pem;
ssl_certificate_key  cert/214225718810040.key;
ssl_session_timeout 5m;
ssl_ciphers ECDHE-RSA-AES128-GCM-
SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH
:!RC4;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
```

具体位置如下：

```
server {
    listen 443;
    server_name  _;
    ssl on;
    root html;
    index index.html index.htm;
    ssl_certificate   cert/214225718810040.pem;
    ssl_certificate_key  cert/214225718810040.key;
    ssl_session_timeout 5m;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        proxy_pass_header Server;
        proxy_set_header Host $http_host;
        proxy_redirect false;
```

重启 Nginx

配置完成后，需要重启 Nginx，在服务器上直接输入如下命令重启 Nginx：

```
service nginx stop
service nginx start
```

至此，已完成服务器端 HTTPS 的准备，此时从客户端使用 HTTPS 请求，就可以保证数据的安全性。

小结

本小节介绍了 HTTPS 的原理及在 Nginx 上的配置和使用方法。作为可选章节，读者在有条件的时候练习即可。