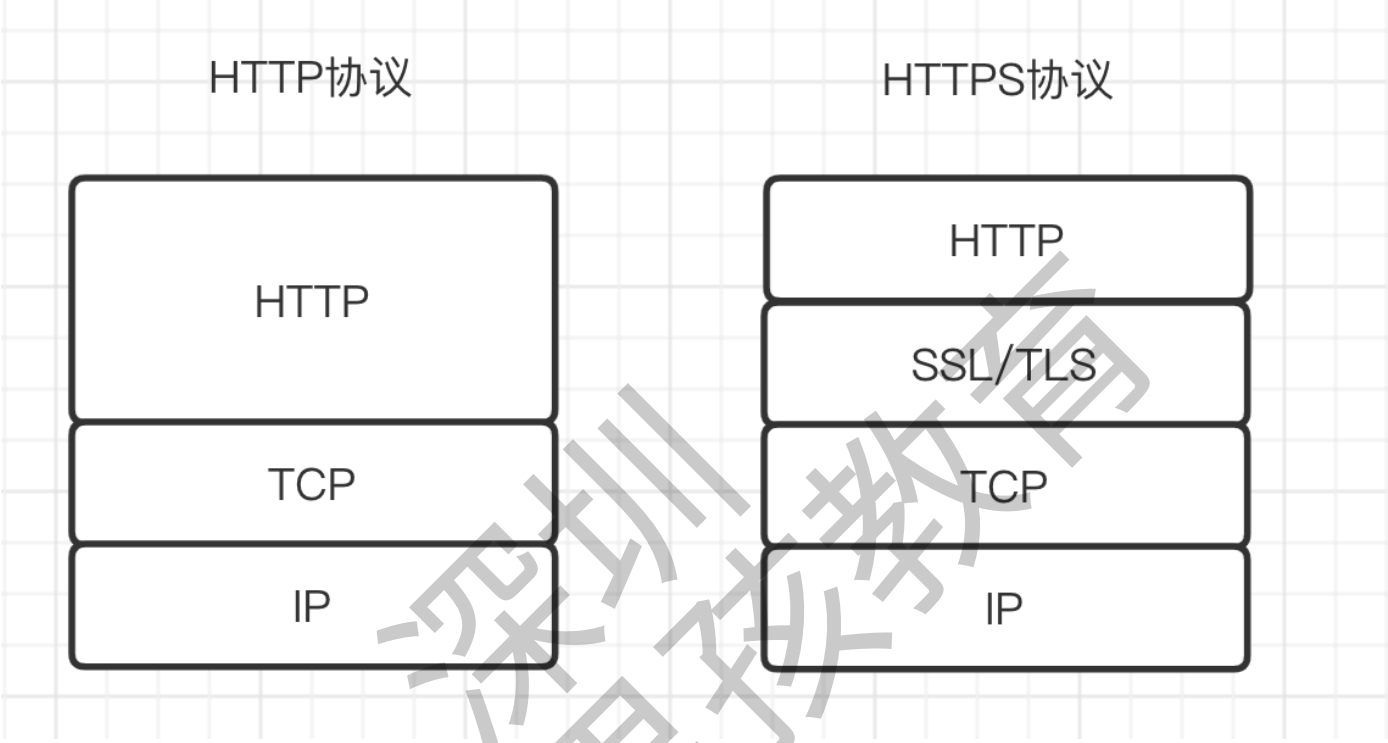
第1章 HTTPS介绍

1.什么是HTTPS协议

```
HTTPS协议不是一个独立的协议，而是HTTP+SSL(安全传输层协议)或HTTP+TLS(安全套接层)协议的组合。
简单来说就是使用SSL或TLS协议对HTTP协议发送的文本进行加密。
```

示意图：



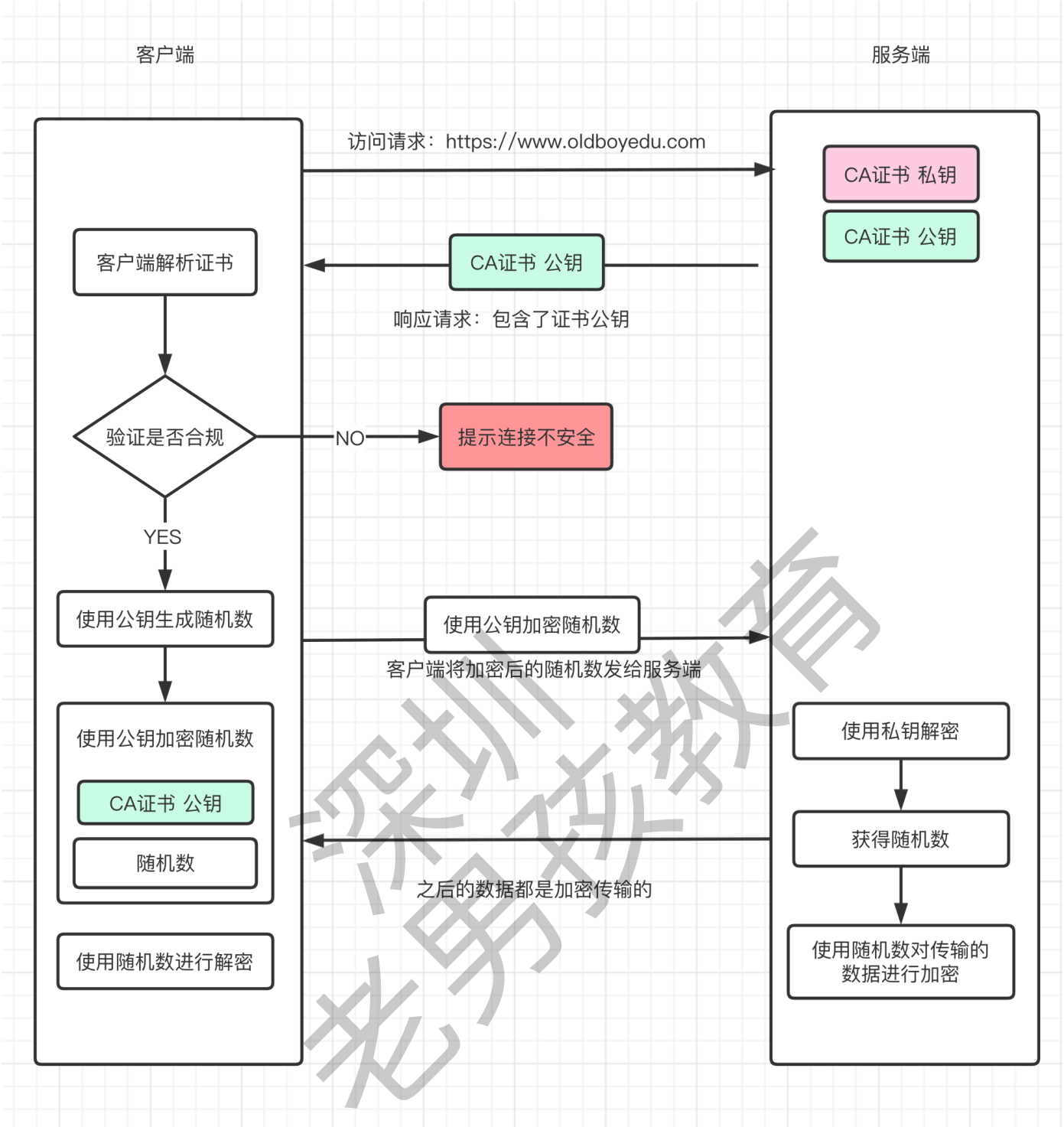
2.为什么需要HTTPS

```
为什么需要使用HTTPS，因为HTTP不安全。当我们使用http网站时，经常会遇到包遭到劫持和篡改，如果采用https协议，那么数据在传输过程中是加密的，所以黑客无法窃取或者篡改数据报文信息。
```

3.HTTP加密流程

文字描述：

```
1.服务端配置有一对数字证书，包含了私钥和公钥，也称为CA证书，CA证书有专门的颁发机构，也可以自己创建，但是自己创建的证书会被浏览器认为不合法。
2.客户端发起HTTPS请求,请求的端口是443
3.服务端接收到请求后会自动将自己的CA证书分发给客户端
4.客户端收到证书后浏览器会做验证，如果是合法的就会显示https图标，如果是不合法的，则会提示连接不安全
5.客户端如果证书通过验证，就会生成一个随机数，然后使用公钥对随机数进行加密
6.然后客户端会将随机数发给服务端
7.服务端收到后使用私钥解密随机数，以后通讯就会使用随机数进行数据加密了
```



第2章 HTTPS购买以及注意事项

1.购买平台

各大云厂商

2.国际常见的证书颁发机构

```
GlobalSign
DigiCert
GeoTrust
```

2.证书类型

```
OV
EV
DV
免费
```

3.域名类型

```
单域名证书    www.mysun.com
多域名证书    www.mysun.com  bbs.mysun.com blog.mysun.com
通配符域名    *.mysun.com
```

4.域名证书购买注意

```
1.一个通配符证书只支持2级域名
2.域名证书最多只能买2年，不支持续费，到期只能买新的
3.域名证书到期后浏览器会提示不安全警告
4.微信小程序必须要求配置https，不然审核不通过
5.提前买
```

5.工作中选择域名的过程

```
1.先收集好所有的域名
2.过滤分析一共有几种类型的域名
  *.www.mysun.com
  *.mysun.com
```

6.如何监控域名过期时间

```
out_time=$(echo | openssl s_client -connect www.oldboyedu.com:443 2>/dev/null |
openssl x509 -noout -dates|awk -F"=" '/notAfter/{print $2}')
out_time_unix=$(date +%s -d "${out_time}")
today=$(date +%s)
let out_day=($out_time_unix-$today)/86400
echo $out_day
```

第3章 简单nginx配置https

1.检查Nginx是否有SSL模块

```
nginx -V
--with-http_ssl_module
```

2.创建证书目录并生成证书

```
mkdir /etc/nginx/ssl_key
cd /etc/nginx/ssl_key
openssl genrsa -idea -out server.key 2048
```

3.生成自签证书，同时去掉私钥的密码

```
openssl req -days 36500 -x509 -sha256 -nodes -newkey rsa:2048 -keyout server.key -out server.crt
CN
SH
SH
mysun
SA
mysun
mysun@qq.com
```

4.创建nginx配置文件

```
cat >/etc/nginx/conf.d/ssl.conf <<EOF
server {
    listen 443 ssl;
    server_name ssl.mysun.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
        root /code;
        index index.html;
    }
}
EOF
```

5.测试重启Nginx

```
nginx -t
systemctl restart nginx
```

6.写入测试文件

```
echo "$(hostname) SSL" > /code/index.html
```

7.访问测试

```
[root@web-7 /etc/nginx/conf.d]# cat ssl.conf
server {
    listen 80;
    server_name ssl.mysun.com;
    rewrite ^(.*) https://$server_name$1 redirect;
}

server {
    listen 443 ssl;
    server_name ssl.mysun.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
        root /code;
        index index.html;
    }
}
```

第4章 强制http跳转到https

1.配置nginx配置文件

```
[root@web-7 /etc/nginx/conf.d]# cat ssl.conf
server {
    listen 80;
    server_name ssl.mysun.com;
    rewrite ^(.*) https://$server_name$1 redirect;
}

server {
    listen 443 ssl;
    server_name ssl.mysun.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
        root /code;
        index index.html;
    }
}
```

第5章 Nginx集群配置https

1.复制已经创建好的证书到其他的web服务器

```
cd /etc/nginx/
scp -r ssl_key 10.0.0.8:/etc/nginx/
scp -r conf.d/ssl.conf 10.0.0.8:/etc/nginx/conf.d/
echo "$(hostname) SSL" > /code/index.html
```

2.复制已经创建好的证书到lb服务器

```
cd /etc/nginx/
scp -r ssl_key 10.0.0.5:/etc/nginx/
```

3.第一种情况：

lb服务器http强制跳转https

lb服务器配置：

```
[root@lb-5 /etc/nginx/conf.d]# cat ssl.conf
upstream ssl_pools {
   server 172.16.1.7:443;
   server 172.16.1.8:443;
}

server {
   listen 80;
   server_name  ssl.oldboy.com ;
   rewrite ^(.*) https://$server_name$1 redirect;
}

server {
    listen 443 ssl;
    server_name ssl.oldboy.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
      proxy_pass  https://ssl_pools;
      include proxy_params;
    }
}
```

web服务器配置：

```
[root@web-8 /etc/nginx/conf.d]# cat ssl.conf
server {
    listen 443 ssl;
    server_name ssl.oldboy.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
        root /code;
        index index.html;
    }
}
```

4.第二种情况：

lb服务器负责https加解密，后端web服务器还是80端口
1.lb服务器配置

```
[root@lb-5 ~]# cat /etc/nginx/conf.d/ssl.conf
upstream ssl_pools {
   server 172.16.1.7;
   server 172.16.1.8;
}

server {
   listen 80;
   server_name  ssl.oldboy.com ;
   rewrite ^(.*) https://$server_name$1 redirect;
}

server {
    listen 443 ssl;
    server_name ssl.oldboy.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
      proxy_pass  http://ssl_pools;
      include proxy_params;
    }
}
```

2.web服务器配置

```
[root@web-7 /etc/nginx/conf.d]# cat ssl.conf
server {
    listen 80;
    server_name ssl.oldboy.com;
    location / {
        root /code;
        index index.html;
    }
}
```

第6章 wordpress配置https

1.lb服务器配置

1.配置nginx配置文件

```
[root@lb-5 ~]# cat /etc/nginx/conf.d/ssl.conf
upstream ssl_pools {
   server 172.16.1.7;
   server 172.16.1.8;
}

server {
   listen 80;
   server_name  blog.mysun.com;
   rewrite ^(.*) https://$server_name$1 redirect;
}

server {
    listen 443 ssl;
    server_name blog.mysun.com;
    ssl_certificate ssl_key/server.crt;
    ssl_certificate_key ssl_key/server.key;
    location / {
      proxy_pass  http://ssl_pools;
      include proxy_params;
    }
}
```

2.web服务器配置

2台web服务器都需要配置

1.配置fastcgi的https相关参数

```
echo "fastcgi_param  HTTPS on;" >> /etc/nginx/fastcgi_params
```

3.web服务器nginx配置

```
[root@web-7 ~]# cat /etc/nginx/conf.d/blog.conf
server {
    listen 80;
    server_name blog.mysun.com;
    root /code/wordpress;
    index index.php index.html;

    location ~ \.php$ {
        root /code/wordpress;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }
}
```

4.重启nginx

```
nginx -t
systemctl restart nginx
```

第7章 阿里云配置HTTPS

1.阿里云https申请流程

阿里云的https证书可以申请个人免费版