一、HTTP协议的概述

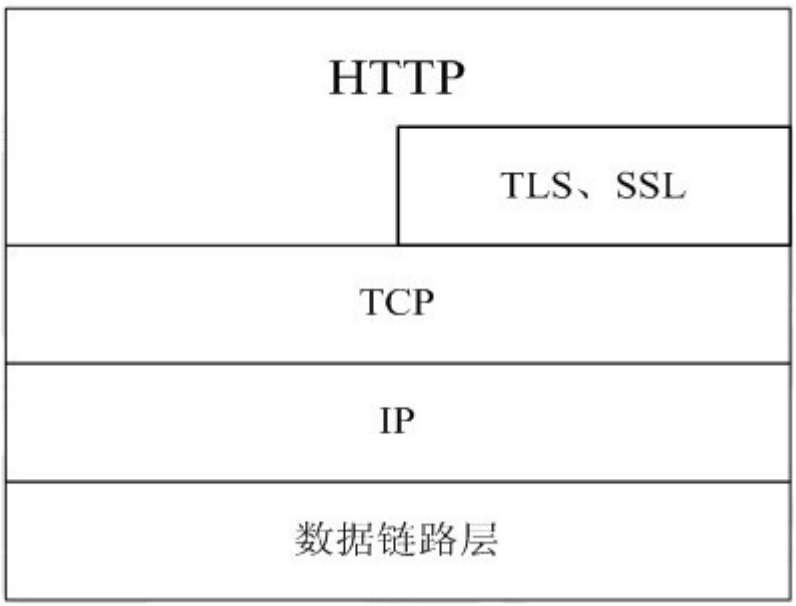
HTTP是Hyper Text Transfer Protocol（超文本传输协议）的缩写。它的发展是万维网协会（World Wide Web Consortium）和Internet工作小组IETF（Internet Engineering Task Force）合作的结果，（他们）最终发布了一系列的RFC，RFC 1945定义了HTTP/1.0版本。其中最著名的就是RFC 2616。RFC 2616定义了今天普遍使用的一个版本——HTTP 1.1。

HTTP协议（HyperText Transfer Protocol，超文本传输协议）是用于从WWW服务器传输超文本到本地浏览器的传送协议。它可以使浏览器更加高效，使网络传输减少。它不仅保证计算机正确快速地传输超文本文档，还确定传输文档中的哪一部分，以及哪部分内容首先显示(如文本先于图形)等。

HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议。

1、在TCP/IP协议栈中的位置

HTTP协议通常承载于TCP协议之上，有时也承载于TLS或SSL协议层之上，这个时候，就成了我们常说的HTTPS。如下图所示：



默认HTTP的端口号为80，HTTPS的端口号为443。

1.2、HTTP的请求响应模型

HTTP协议永远都是客户端发起请求，服务器回送响应。见下图：

这样就限制了使用HTTP协议，无法实现在客户端没有发起请求的时候，服务器将消息推送给客户端。

HTTP协议是一个无状态的协议，同一个客户端的这次请求和上次请求是没有对应关系。

1.3、工作流程

一次http操作称为一个事务，其工作过程可分为四步：

1）首先客户端与服务器需要建立连接，只需要单击某个超链接，http的工作开始。

2）建立连接后客户机发送一个连接给服务器，请求方式的个格式为：统一资源标识（URL）、协议版本号、后边是MIME信息包括请求修饰符

客户机信息和可能的内容

3）服务器接到请求之后，给予相应的响应信息，其格式为一个状态行，包括信息的协议版本号、一个成功或错误的代码、后边是MIME信息包括服务器信息

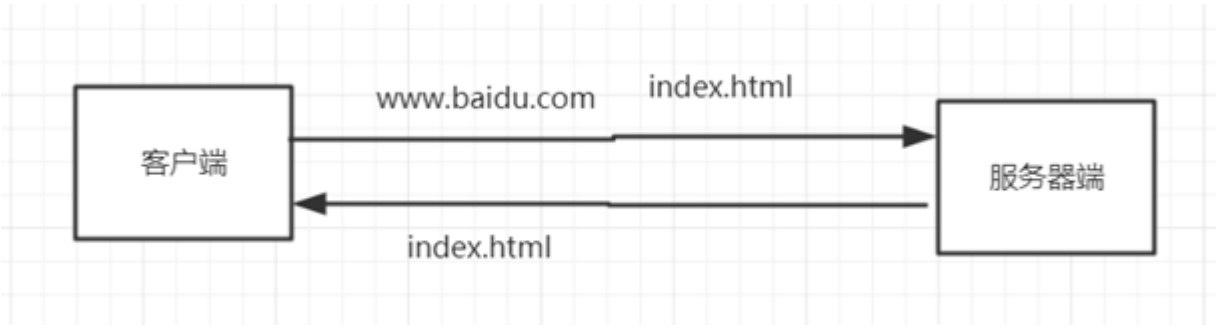
实体信息和可能的内容。

4）客户机接收服务器返回的信息通过浏览器显示在用户的显示屏上，然后客户机与服务器断开连接。

如果以上过程中的某一步出现错误，那么将产生的错误信息返回给客户端

1.4、一些概念

1）客户端与服务端



2） 资源

html/文本、word、avi电影、其他资源

3） 媒体类型

MIME类型。 text/html、 image/jpeg

4） URI和URL

URI:web服务器资源的名字。 index.html

http://www.gupaoedu.com:80/java/index.html[?query-string] #location

schema: http/https/ftp.

host: web服务器的ip地址或者域名

port: 服务端端口， http默认访问的端口是80

path: 资源访问路径

query-string: 查询参数

5） 方法

GET/PUT/DELETE/POST/HEAD

1.5、报文

request参数、 response响应参数

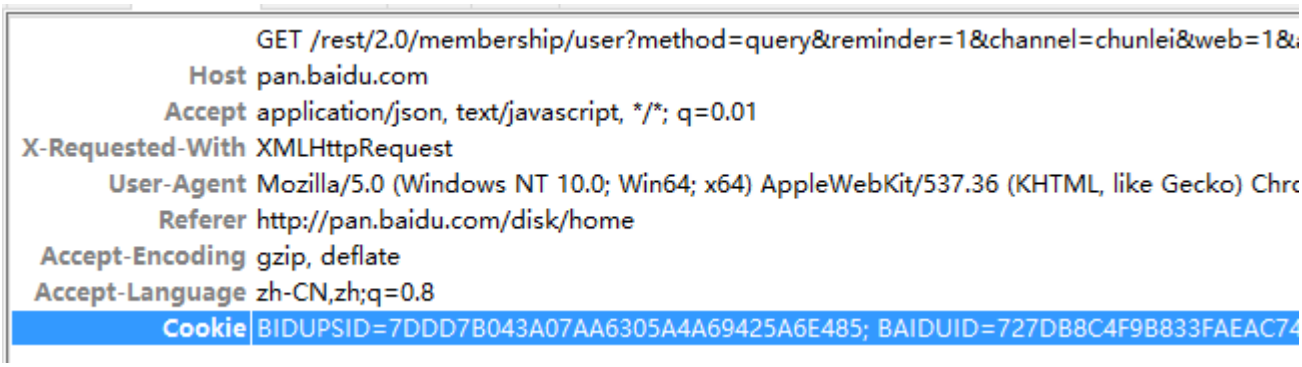
request消息结构包含三部分： （起始行、首部字段、主体）

METHOD /path / http/version-number

Header-Name:value

空行

主体 optional request body



response

http/version-number status code message

header-name:value

body

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Content-Length: 99
Content-Type: application/json; charset=UTF-8
Date: Wed, 26 Jul 2017 12:47:23 GMT
Flow-Level: 3
Server: nginx
X-Powered-By: BaiduCloud
Yld: 186126124455703594
Yme: ZIGW/i4rX04SdTYCUmr/tGtPovoeTxz2owpHwyiE
Proxy-Connection: Keep-alive

{"errno":0,"request_id":4797812156975952938,"uinfo":"4T6gtrOsMsL1xyva1fi5sUqj7QRuURZfzBnnMuiiJ34="}
```

演示：使用Wireshark抓TCP、http包

1.6、状态码

http/1.1版本的协议里面定义了五种类型的状态码

1XX 提示信息

2XX 成功

3XX 重定向

4XX 客户端错误

5XX 服务器端的错误

1.7、缓存

HTTP协议的特点

1. 无状态

cookie+session

2. 多次请求
3. 基于TCP协议

1）什么是web缓存？

WEB缓存(cache)位于Web服务器和客户端之间。

缓存会根据请求保存输出内容的副本，例如html页面，图片，文件，当下一个请求来到的时候：如果是相同的URL，缓存直接使用副本响应访问请求，而不是向源服务器再次发送请求。

HTTP协议定义了相关的消息头来使WEB缓存尽可能好的工作。

2）缓存的优点

减少相应延迟：因为请求从缓存服务器(离客户端更近)而不是源服务器被相应，这个过程耗时更少，让web服务器看上去相应更快。

减少网络带宽消耗：当副本被重用时会减低客户端的带宽消耗;客户可以节省带宽费用，控制带宽的需求的增长并更易于管理。

3）缓存相关的HTTP扩展消息头

- expires:提示缓存响应过期时间，格林威治时间GMT
- Cache-Control：更细致的控制缓存的内容
- Last-Modified:响应中资源最后一次修改的时间
- ETag：响应中资源的检验值，在服务器上某个时间段是唯一标识的
- Date:服务器的时间
- If-Modified-Since：客户端存取的该资源最后一次修改的时间，同Last-Modified。
- If-None-Match：客户端存取的该资源的检验值，同ETag。

4）客户端缓存生效的常见流程

服务器收到请求时会在200OK中返回该资源的Last-Modified和ETag头，客户端将该资源保存在cache中，并记录着两个属性

当客户端需要发送相同请求时，会在请求中携带If-Modified-Since和If-None-Match这两个头，这两个头的值分别是响应中的Last-Modified和ETag的值

服务器通过这两个值判断本地资源未发生变化，客户端不需要重新下载，返回304。

常见流程如下图所示：

5）Web缓存的机制

HTTP/1.1中缓存的目的是为了在很多情况下减少发送请求，同时在许多情况下可以不需要发送完整响应。前者减少了网络回路的数量;HTTP利用一个“过期(expiration)”机制来为此目的。后者减少了网络应用的带宽;HTTP用“验证(validation)”机制来为此目的。

HTTP定义了3种缓存机制：

(1)Freshness：允许一个回应消息可以在源服务器不被重新检查，并且可以由服务器和客户端来控制。例如，Expires回应头给了一个文档不可用的时间。Cache-Control中的max-age标识指明了缓存的最长时间;

(2)Validation：用来检查以一个缓存的回应是否仍然可用。例如，如果一个回应有一个Last-Modified回应头，缓存能够使用If-Modified-Since来判断是否已改变，以便判断根据情况发送请求;

(3)Invalidation： 在另一个请求通过缓存的时候，常常有一个副作用。例如，如果一个URL关联到一个缓存回应，但是其后跟着POST、PUT和DELETE的请求的话，缓存就会过期。

6）断点续传和多线程下载的实现原理

HTTP协议的GET方法，支持只请求某个资源的某一部分;

206 Partial Content 部分内容响应;

Range 请求的资源范围;

Content-Range 响应的资源范围;

在连接断开重连时，客户端只请求该资源未下载的部分，而不是重新请求整个资源，来实现断点续传。

分块请求资源实例：

Eg1：Range: bytes=306302- ：请求这个资源从306302个字节到末尾的部分;

Eg2：Content-Range: bytes 306302-604047/604048：响应中指示携带的是该资源的第306302-604047的字节，该资源共604048个字节;

客户端通过并发的请求相同资源的不同片段，来实现对某个资源的并发分块下载。从而达到快速下载的目的。目前流行的FlashGet和迅雷基本都是这个原理。

多线程下载的原理：

下载工具开启多个发出HTTP请求的线程;

每个http请求只请求资源文件的一部分：Content-Range: bytes 20000-40000/47000;

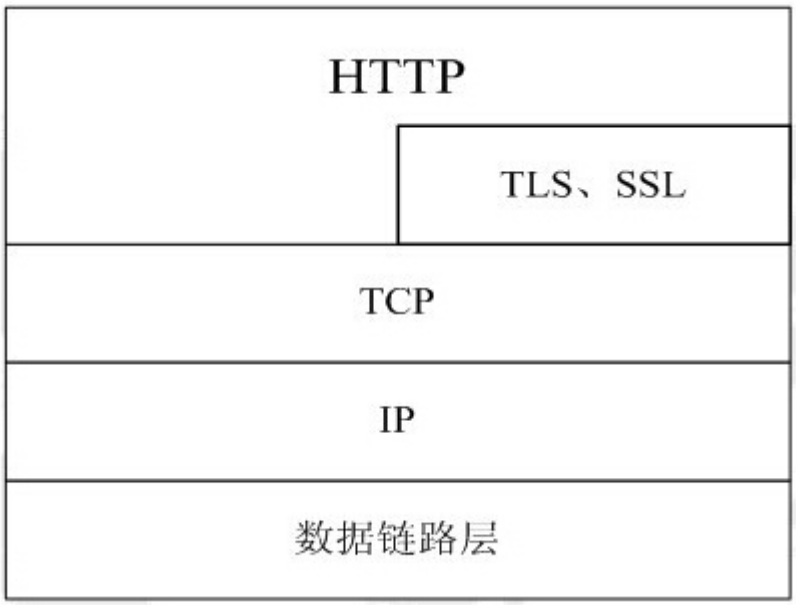
合并每个线程下载的文件。

二、HTTPS通信

1、什么是https?

HTTPS(全称：Hypertext Transfer Protocol over Secure Socket Layer)，是以安全为目标的HTTP通道，简单讲是HTTP的安全版。即HTTP下加入SSL层，HTTPS的安全基础是SSL，因此加密的详细内容请看SSL。

见下图：

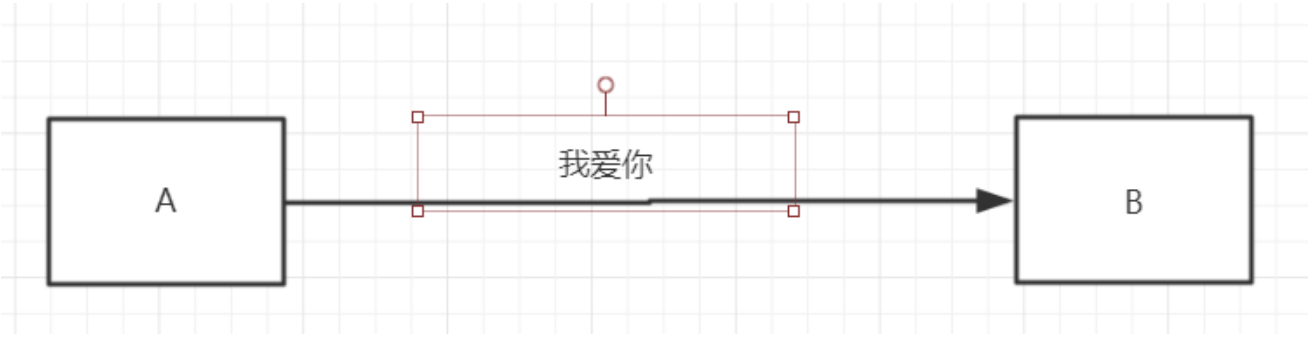


SSL/TLS

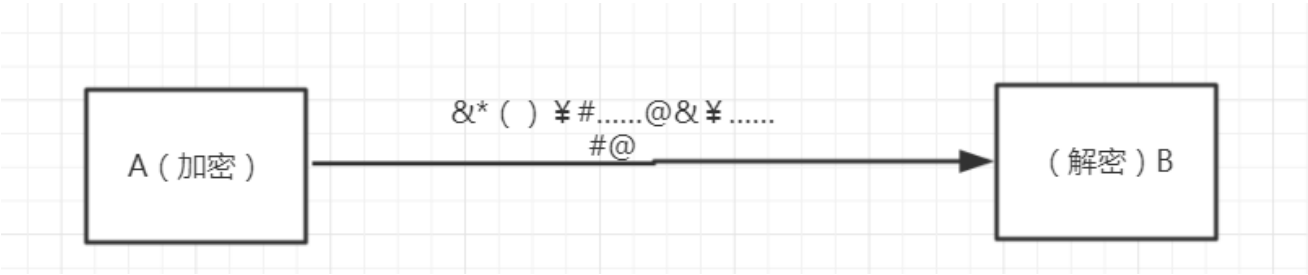
SSL3.0

ISOC 在SSL的基础上发布了升级版本 TLS1.2

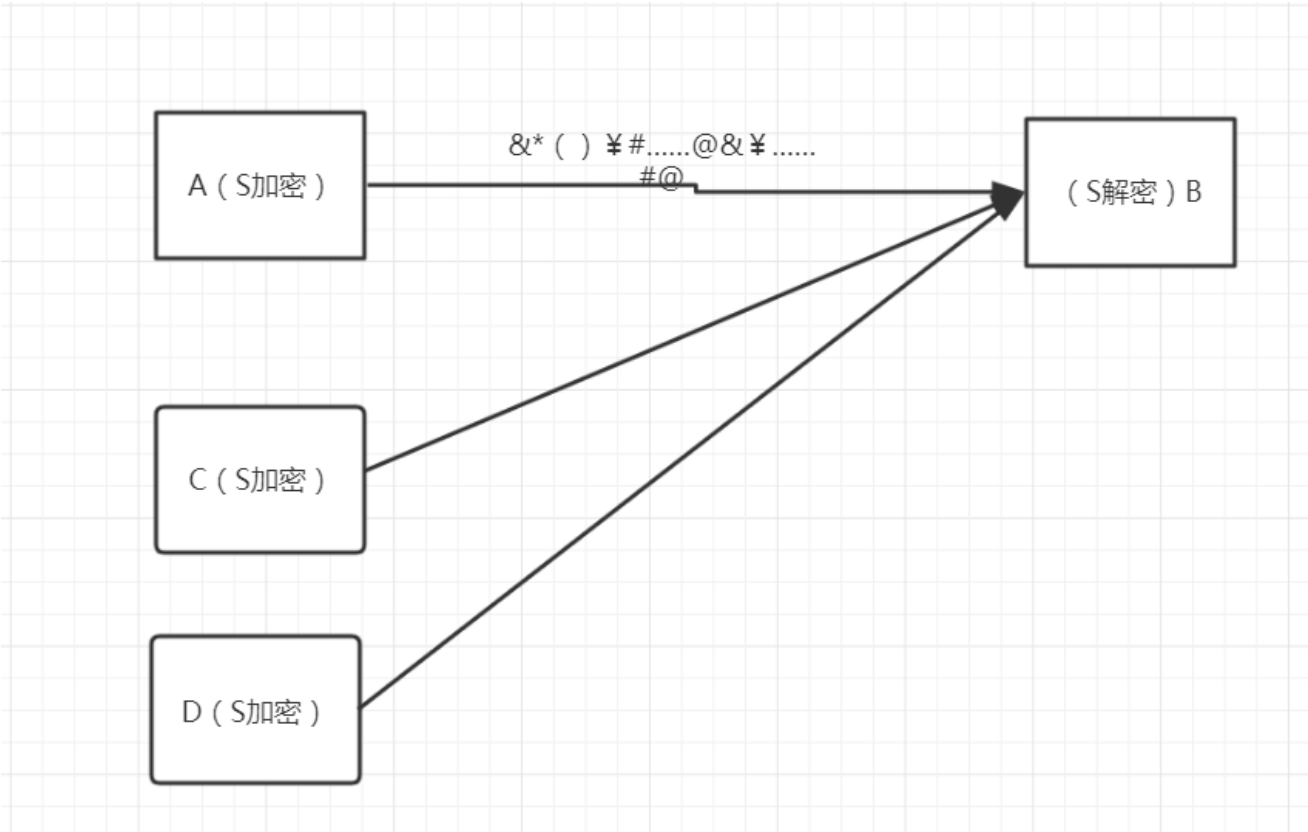
2、HTTPS的工作原理



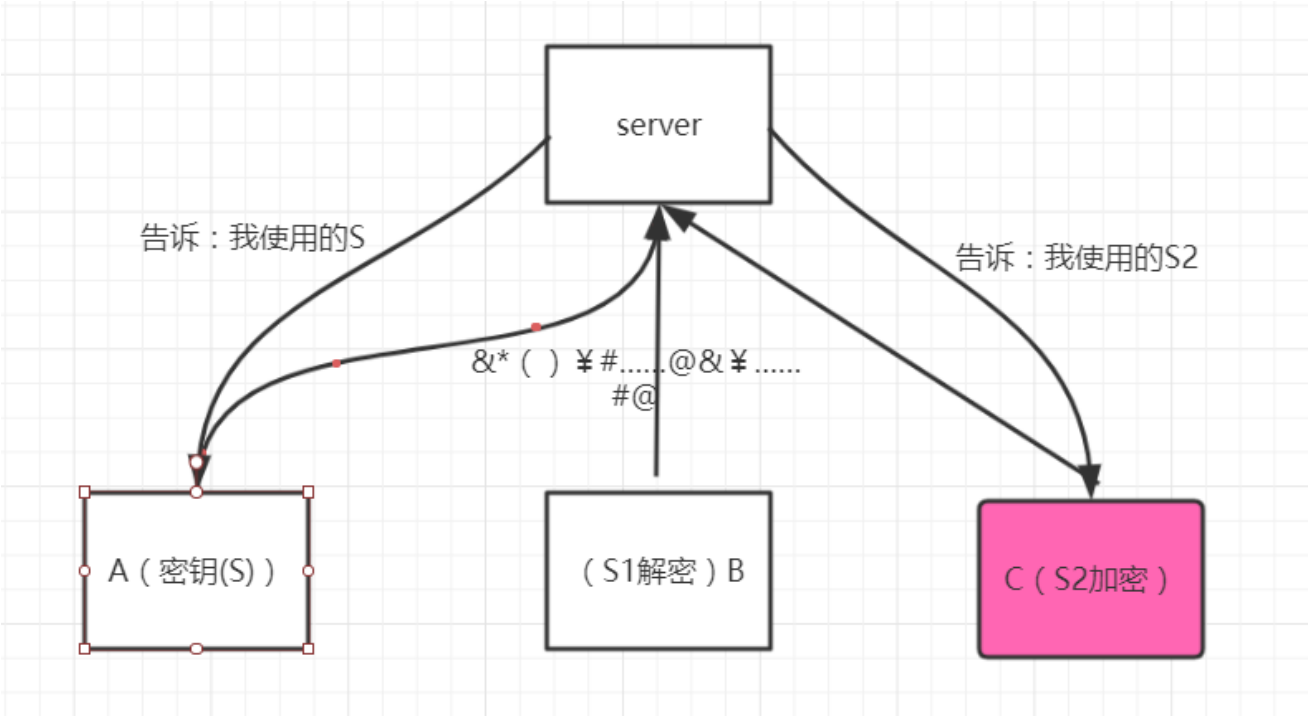
第一步， 使用对称加解密



第二步，密钥是公开的，所有的客户端都可以拿到



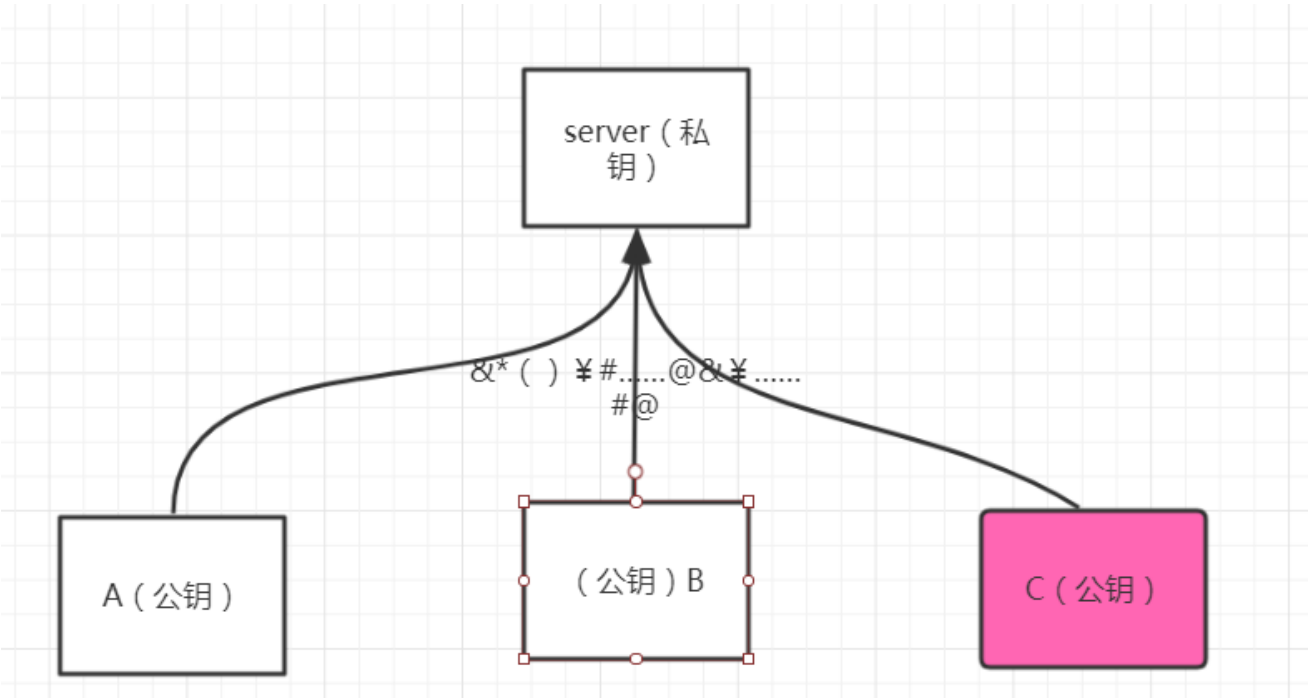
第三步 针对不同的客户端使用不同的密钥



问题：协商过程是没有加密的，所以还会出现被截断的问题

第四步：使用非对称加密

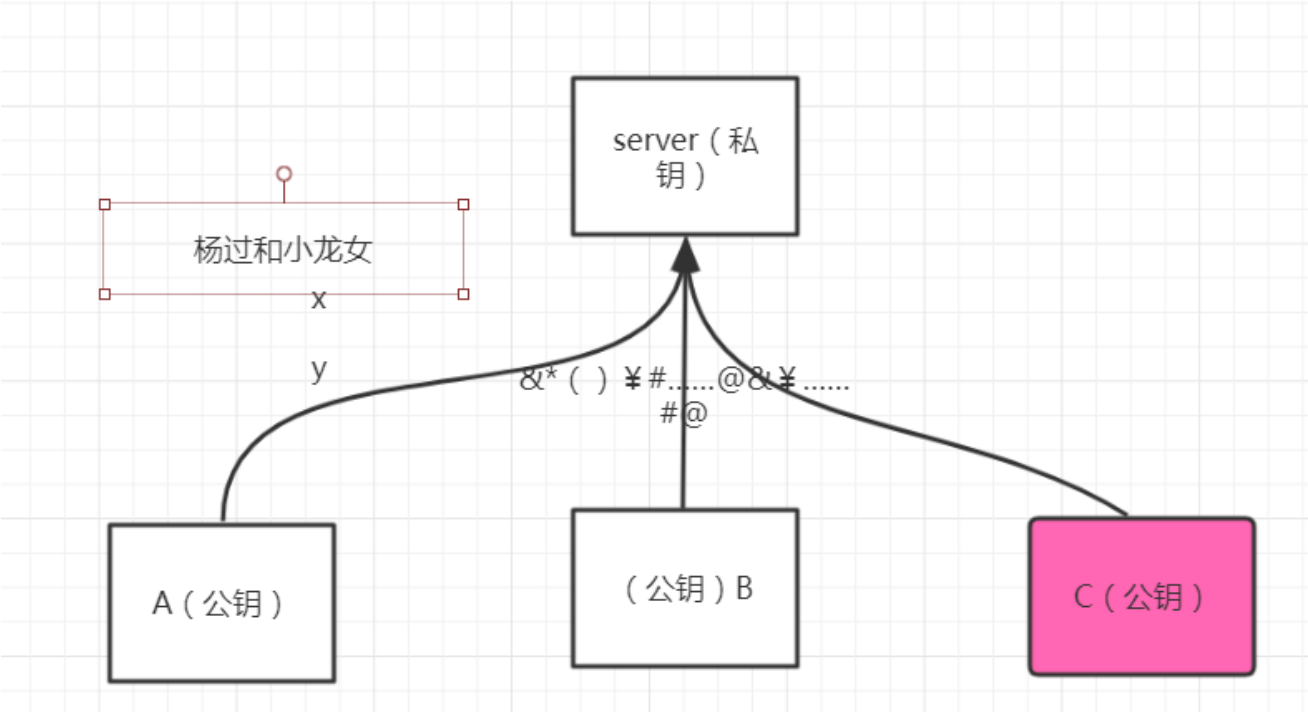
非对称：公钥和私钥的概念



问题： 客户端如何拿到公钥

1. 服务器端把公钥发送给每一个客户端
2. 服务器端把公钥放到远程服务器，客户端可以请求到
3. 让浏览器保存所有的公钥（不现实）

第五步 公钥被调包的问题按照上面的方案，永远存在。



第六步：使用第三方机构来解决

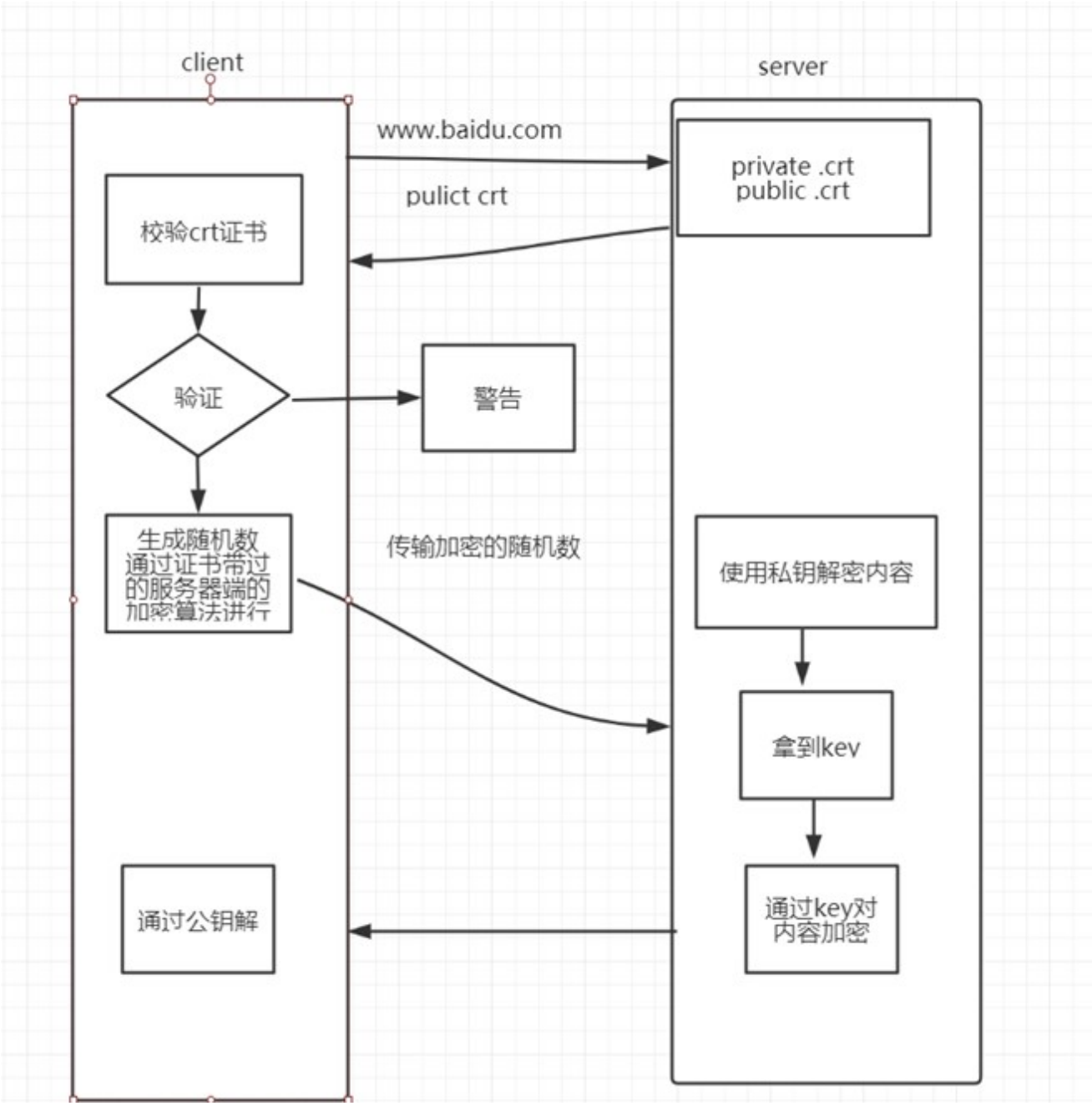
通过第三方机构，使用第三方机构的私钥对我们【需要传输的公钥】进行加密

第七部分

数字证里面包含的内容：

公司信息、网站信息、数字证书的算法、公钥

连接过程


client
server
www.baidu.com
private .crt
public .crt
pulict crt
校验crt证书
验证
警告
生成随机数
通过证书带过
的服务器端的
加密算法进行
传输加密的随机数
使用私钥解密内容
拿到kev
通过key对
内容加密
通过公钥解