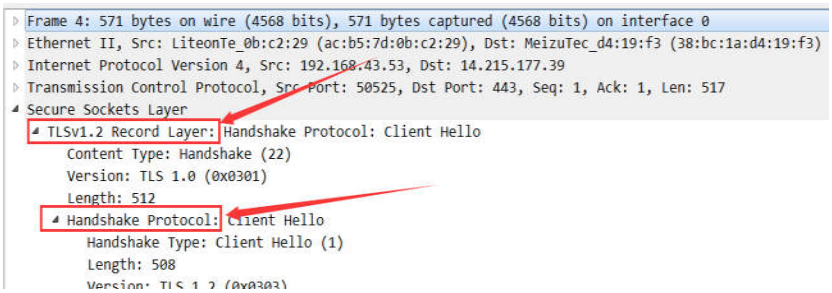
网络安全协议

一．TLS 协议概述

TLS：安全传输层协议，用于两个通信应用程序之间提供保密性和数据完整性，该协议由两层组成：TLS 记录协议（TLS Record）和 TLS 握手协议（TLS Handshake）。



先简单看一下记录协议的内容：

TLS Record：由上图可知，TLS Record 是位 TLS Handshake 上方，可以看到其结构为：

1.Content Type(1byte)

Recode 类型主要有下面几种：

change_cipher_spec(20)；

alert(21)；

handshake(22)；

application_data(23)；

2.Version(2bytes)

TLS 的版本号，0301 表示 TLS1.0,0302 表示 TLS1.1,0303 表示 TLS1.2;

3.Length(2bytes)

Body 数据的长度，最大为 2^14;

4.Body

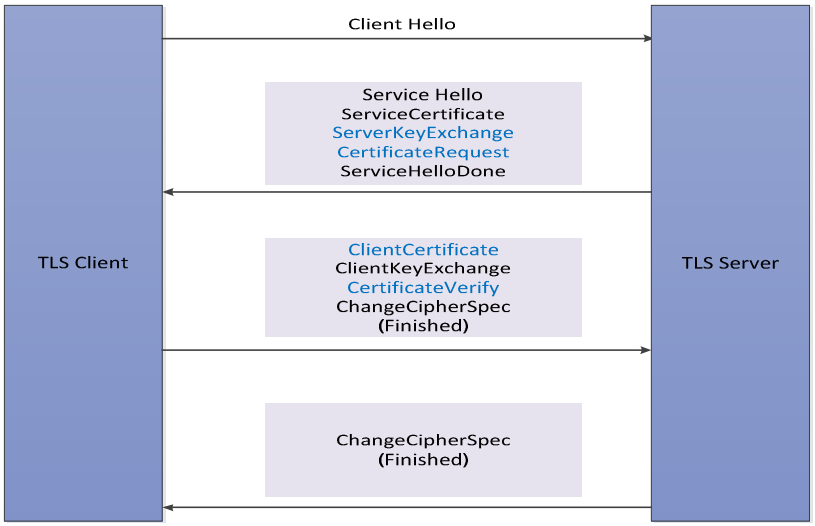
消息体，具体格式由 ContentType、是否压缩、是否加密决定

下面主要对握手协议进行分析；

Handshark Protocol 如下所示：

```
Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 508
    Version: TLS 1.2 (0x0303)
  ▷ Random
    Session ID Length: 32
    Session ID: 80882b3d5cd53ce01fb9493a6ab2f3955483fbe4a5ef9de3...
    Cipher Suites Length: 30
  ▷ Cipher Suites (15 suites)
    Compression Methods Length: 1
  ▷ Compression Methods (1 method)
    Extensions Length: 405
  ▷ Extension: server_name
  ▷ Extension: Extended Master Secret
  ▷ Extension: renegotiation_info
  ▷ Extension: elliptic_curves
  ▷ Extension: ec_point_formats
  ▷ Extension: SessionTicket TLS
  ▷ Extension: Application Layer Protocol Negotiation
  ▷ Extension: status_request
  ▷ Extension: signature_algorithms
  ▷ Extension: Padding
```

HandsharkProtocal:握手协议，其主要流程图如下所示：



第一阶段：

1.声明支持的协议版本；
2.声明支持的加密算法；
3.声明 Session ID；
4.客户端生成随机数；

其结构为：

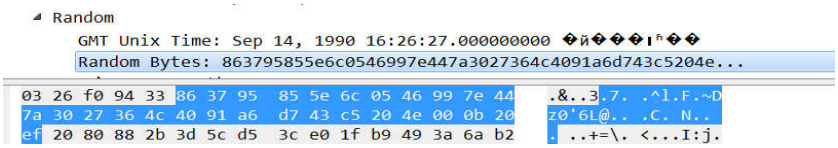
1. Handshark Type：Client Hello；

```
Handshake Protocol: Client Hello
    Handshake Type: Client Hello (1)
    Length: 508
    Version: TLS 1.2 (0x0303)
```

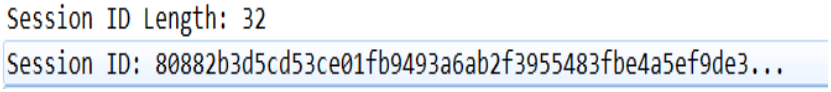
2.Length:508　消息长度为 508bytes ；

3.版本号：
客户端所支持的最高 SSL 版本，0301 表示 TLS1.0,0302 表示 TLS1.1,0303 表示 TLS1.2;

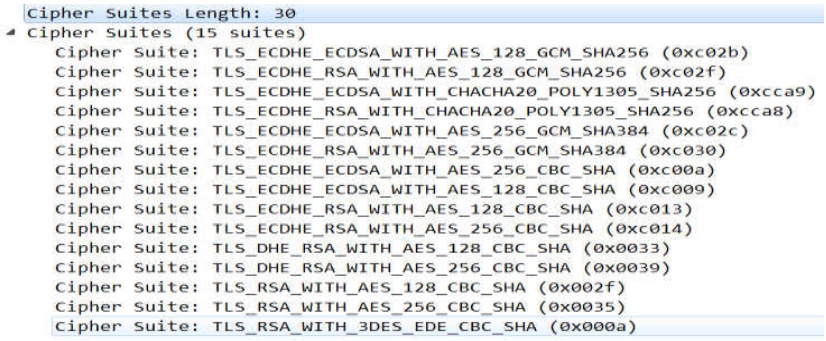
4. 随机数：由客户端生成的随机数，用 32 位时间戳和一个安全随机数生成器生成的 28 字节随机数组成；



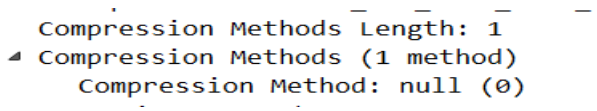
5. 会话 ID：非 0 值代表客户端想更新现有连接参数，0 值表示客户端想在新会话上创建一个新连接；



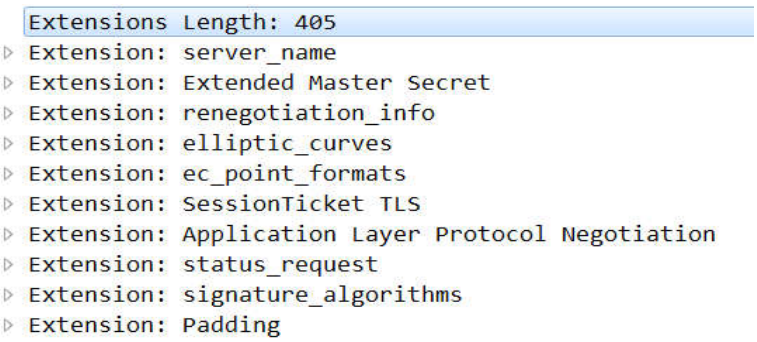
6. 密码组：按优先级的降序排列、客户端支持的密码算法列表。表的每个元素定义了一个密钥交换算法和一个密码说明；



7. 压缩方法：图中消息只支持一种压缩方法—NULL，该方法不会产生任何压缩结果；

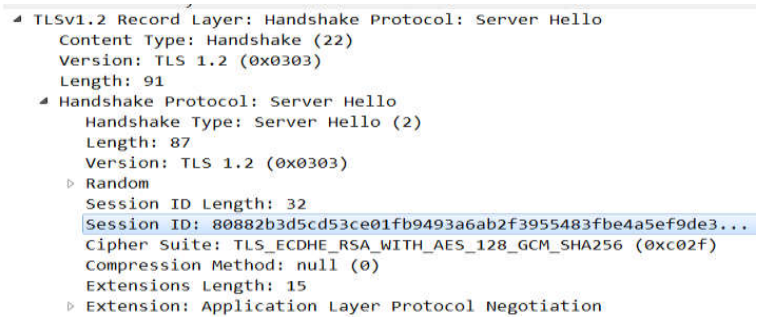


8. 扩展数据：图中有 405 个字节用于扩展，主要包含一些支持的椭圆参数以及签名方式；

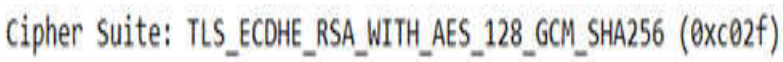


第二阶段：

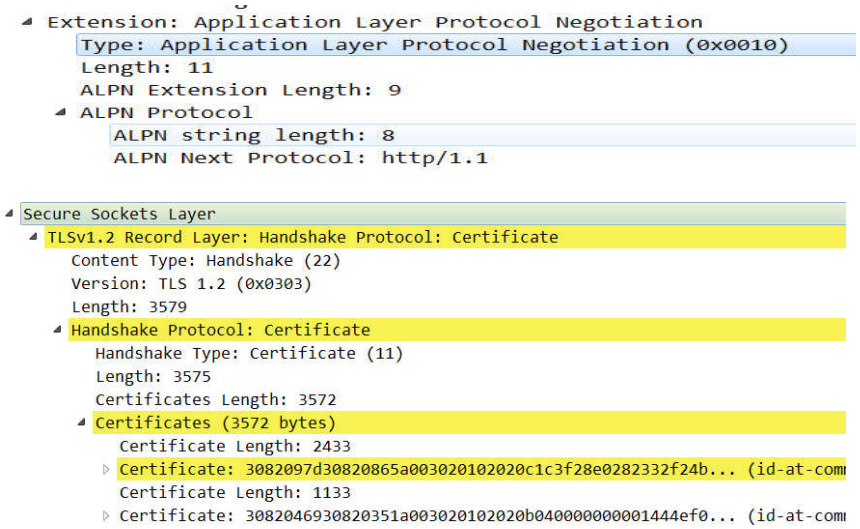
1.确认使用的协议版本；
2.服务器生成的随机数；
3.确认 Session ID
4.确认使用的加密算法
5.发送服务器证书
6.服务器密钥交换，即发送服务器端公钥（非必须）
7.要求客户端提供证书（非必须）



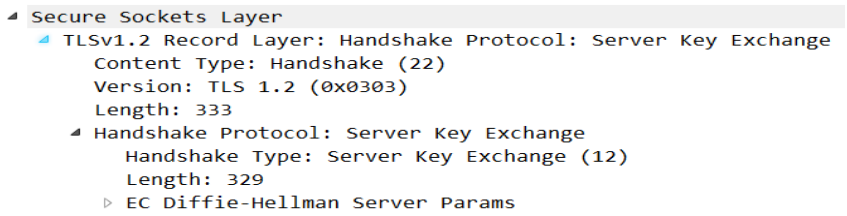
1.Handshark Type：Server Hello；
2.Length:消息长度为 87 个字节；
3.Version:TLS 1.2；
4.Random：由服务器生成，与客户端随机数相互独立；
5.Session:如果客户端会话数值非 0，则直接使用客户端数值，如果客户端数值为 0，则服务器提供 32 字节的会话 ID；
6.Cipher Suite:从客户端提供的密码组中挑选出来的密码组；



7.压缩方法：从客户端提供的压缩方法中挑选出来的压缩方法即：NULL
8.扩展数据：Application Layer Protocol Negotiation（ALPN），在此处主要用户协商 HTTP 协议版本；

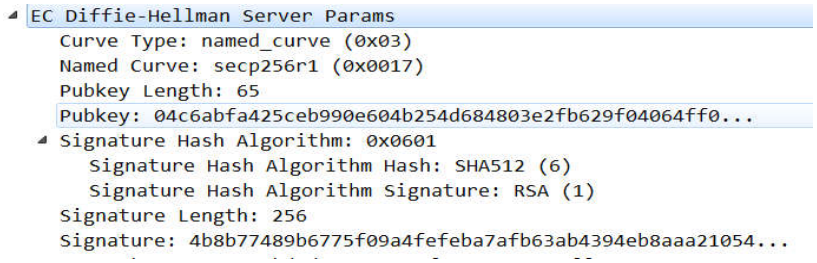


上图可以看到 Certificate 内包含两个证书，一个是百度服务器证书，一个是 CA 机构的证书，服务器将一组（两个）证书下发给客户端；



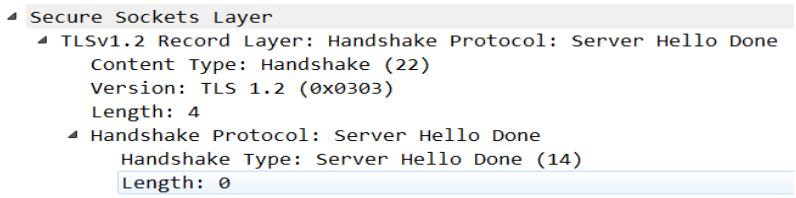
上图是密钥交换消息：

1.EC Diffie-Hellman Server Params，该段描述 ECDH 密钥分发的一些详细信息，如下图：



上图可见，该消息定义 ECDH 的椭圆曲线参数 Name Curve:secp256r1，生成公钥，另外该消息同时生成签名一并发送给客户端。

Server Hello Done:服务器发送数据结束的标志



第三个阶段：

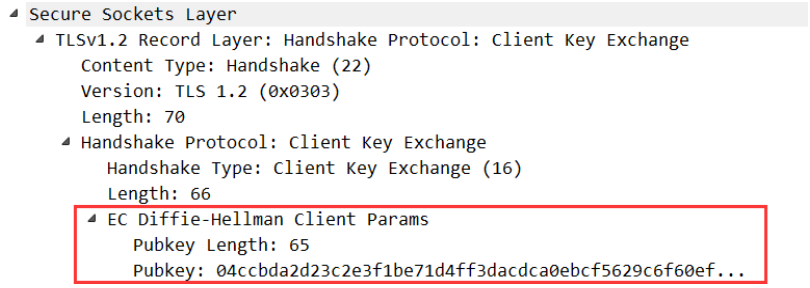
1.客户端发送自己的证书(非必须)

2.Client Key Exchange

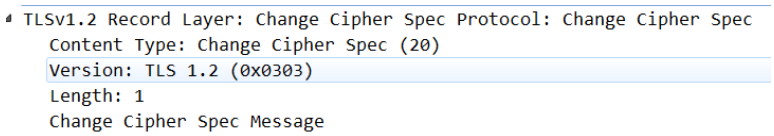
3.Certificate Verify(非必须)

4.Change Cipher Spec

1.客户端生成 ECDH 公钥，并发送给服务器，如下图所示：

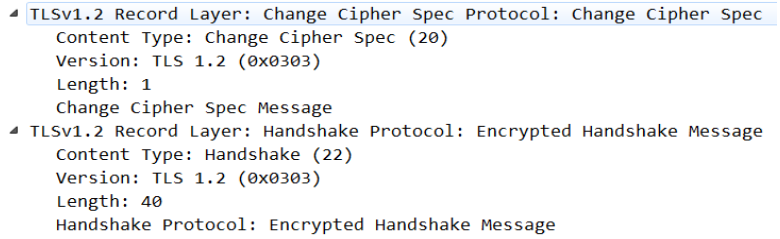


2. Change Cipher Spec：客户端通知服务器开始使用加密方式发送报文，至此第三阶段结束：



第四阶段：

服务器端 Change Cipher Spec



服务器发送 Change Cipher Spec 后，握手到此结束，后续消息使用加密算法对消息进行加密；

二．SSL 协议

因 SSL 和 TLS 大致相同，在此不多做论述，仅列出之间差异的部分。
最新版本的 TLS 是建立在 SSL 3.0 协议规范之上，是 SSL 3.0 的后续版本。在 TLS 与 SSL3.0 之间的差别主要体现在它们所支持的加密算法不同。

TLS 与 SSL 的差异：

1）版本号：TLS 记录格式与 SSL 记录格式相同，但版本号的值不同，TLS 的版本 1.0 使用的版本号为 SSLv3.1。

2）报文鉴别码：SSLv3.0 和 TLS 的 MAC 算法及 MAC 计算的范围不同。TLS 使用了 RFC-2104 定义的 HMAC 算法。SSLv3.0 使用了相似的算法，两者差别在于 SSLv3.0 中，填充字节与密钥之间采用的是连接运算，而 HMAC 算法采用的是异或运算。

3）伪随机函数：TLS 使用了称为 PRF 的伪随机函数来将密钥扩展成数据块，是更安全的方式。

4）报警代码：TLS 支持几乎所有的 SSLv3.0 报警代码，而且 TLS 还补充定义了很多报警代码，如解密失败（decryption_failed）、记录溢出（record_overflow）、未知 CA（unknown_ca）、拒绝访问（access_denied）等。

5）密文组和客户证书：SSLv3.0 和 TLS 存在少量差别，即 TLS 不支持 Fortezza 密钥交换、加密算法和客户证书。

6）certificate_verify 和 finished 消息：SSLv3.0 和 TLS 在用 certificate_verify 和 finished 消息计算 MD5 和 SHA-1 散列码时，计算的输入有少许差别，但安全性相当。

7）加密计算：TLS 与 SSLv3.0 在计算主密值（master secret）时采用的方式不同。

8）填充：用户数据加密之前需要增加的填充字节。在 SSL 中，填充后的数据长度要达到密文块长度的最小整数倍。而在 TLS 中，填充后的数据长度可以是密文块长度的任意整数倍（但填充的最大长度为 255 字节），这种方式可以防止基于对报文长度进行分析的攻击。

三．HTTPS 协议

1.HTTPS 协议概念

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，简单讲就是 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就需要 SSL。

2.HTTPS 协议作用

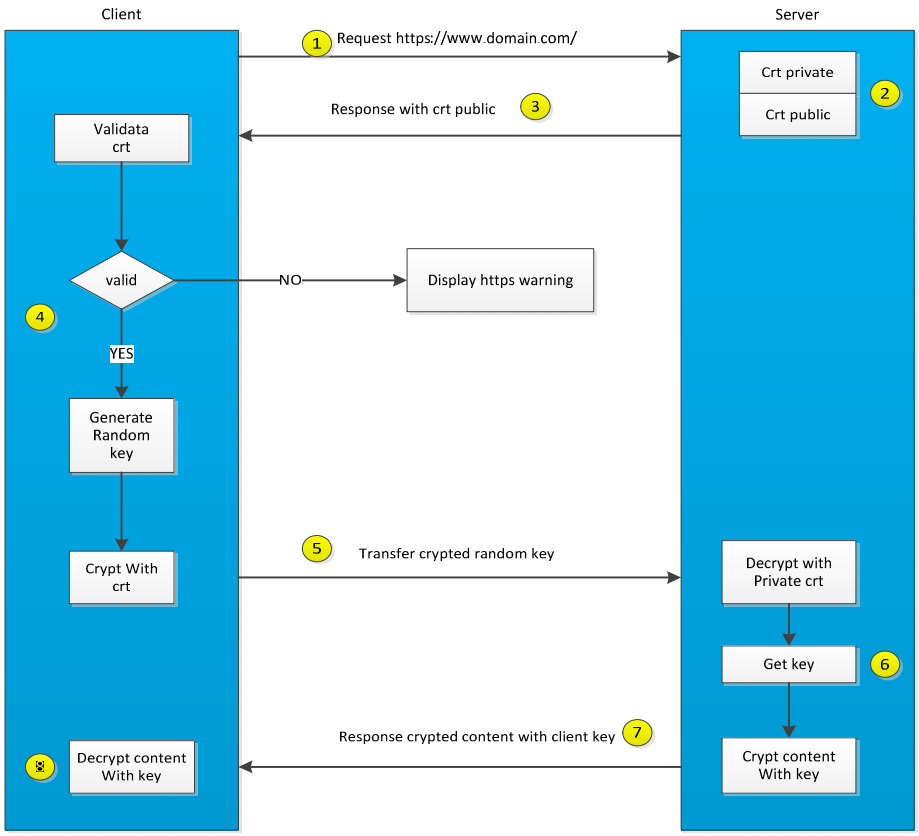
HTTPS 协议的主要作用可以分为两种：

(1). 建立一个信息安全通道，来保证数据传输的安全；

(2). 确认网站的真实性。

3.HTTPS 加密及验证过程

HTTPS 加密及验证过程，如下图所示：



如上图所示：

①. 客户端发起 HTTPS 请求

用户在浏览器里输入一个 https 网址，然后连接到 server 的 443 端口。

②.服务端的配置

采用 HTTPS 协议的服务器必须要有一套数字证书，可以自己制作，也可以向组织申请，区别就是自己颁发的证书需要客户端验证通过，才可以继续访问，而使用受信任的公司申请的证书则不会弹出提示页面,这套证书其实就是一对公钥和私钥。

③.传送证书

这个证书其实就是公钥，只是包含了很多信息，如证书的颁发机构，过期时间等等。

④.客户端解析证书

这部分工作是由客户端的 TLS 来完成的，首先会验证公钥是否有效，比如颁发机构，过期时间等等，如果发现异常，则会弹出一个警告框，提示证书存在问题。如果证书没有问题，那么就生成一个随机值，然后用证书(公钥)对该随机值进行加密。

⑤.传送加密信息

这部分传送的是用证书(公钥)加密后的随机值,目的就是让服务端得到这个随机值，以后客户端和服务端的通信就可以通过这个随机值来进行加密解密了。

⑥.服务段解密信息

服务端用私钥解密后，得到了客户端传过来的随机值(密钥)，然后把内容通过该值进行加密。

⑦.传输加密后的信息

这部分信息是服务端用随机值(密钥)加密后的信息，可以在客户端被还原。

⑧.客户端解密信息

客户端用之前生成的随机值(密钥)解密服务端传过来的信息,于是获取了解密后的内容，整个过程第三方即使监听到了数据，也束手无策。

4.HTTPS 的优点

(1).使用 HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户端和服务器；

(2).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，可防止数据在传输过程中不被窃取、改变，确保数据的完整性；

(3).大幅增加了中间人攻击的成本。

5.HTTPS 的缺点

(1).SSL 证书需要钱，功能越强大的证书费用越高；

(2).SSL 证书通常需要绑定 IP，不能在同一 IP 上绑定多个域名，IPv4 资源不可能支撑这个消耗；

(3).HTTPS 连接缓存不如 HTTP 高效，大流量网站如非必要也不会采用，流量成本太高。
(4).HTTPS 连接服务器端资源占用高很多，支持访客稍多的网站需要投入更大的成本；
(5).HTTPS 协议握手阶段比较费时，对网站的相应速度有负面影响；

6.HTTPS 和 HTTP 的区别

HTTPS 和 HTTP 的区别主要如下：

1. https 协议需要到 ca 申请证书，一般免费证书较少，因而需要一定费用。
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，用的端口也不一样，前者是 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。

四. IPSec 协议

1. IPSec 协议简介

IPSec（IP Security）是 IETF（Internet Engineering Task Force，Internet 工程任务组）制定的 IP 安全协议集。它给出了应用于 IP 层上网络数据安全的一整套体系结构，包括网络认证协议 Authentication Header（AH）、封装安全载荷协议 Encapsulating Security Payload（ESP）、密钥管理协议 Internet Key Exchange （IKE）和用于网络认证及加密的一些算法等。这些协议用于提供数据认证、数据完整性和加密性三种保护形式。其中，IPsec 中的 AH 协议定义了认证的应用方法，提供数据源认证和完整性保证；ESP 协议定义了加密和可选认证的应用方法，提供数据可靠性保证。而 IKE 主要是对密钥进行交换管理，对算法、协议和密钥 3 个方面进行协商。

2. IPSec 作用目标

数据机密性（Confidentiality）：IPSec 发送方在通过网络传输包前对包进行加密。

数据完整性（Data Integrity）：IPsec 接收方对发送方发送来的包进行认证，以确保数据在传输过程中没有被篡改。

数据来源认证（Data Authentication）：IPsec 在接收端可以认证发送 IPsec 报文的发送端是否合法。

防重放（Anti-Replay）：IPsec 接收方可检测并拒绝接收过时或重复的报文。

3. IPSec 通信过程

IPsec 是在逐个包上执行的。当实现 IPSec 时，每个发往外部的 IP 包都会在发送之前有 IPSec 逻辑进行处理，而每个发往内部的 IP 包也会在接收到之后并且在传递给上一层（如 TCP 或 UDP）之前有 IPSec 处理。

向外发包:

例如将 TCP 层的上一层的数据块传递到 IP 层并形成 IP 包，然后执行以下步骤：

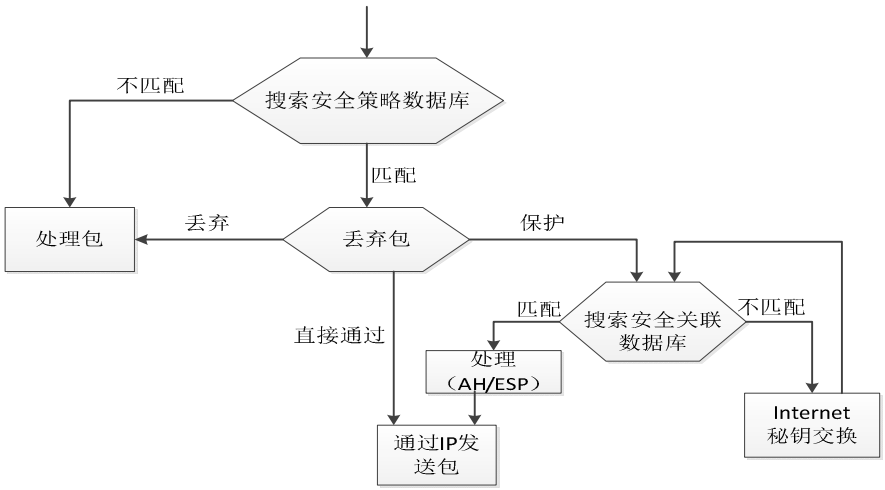
（1） IPSec 搜索与 IP 包匹配的安全策略数据库 SPD（Security Policy Database）。

（2） 假如未找到匹配的 SPD，则丢弃该包并生成错误信息。

（3） 假如找到匹配的 SPD,则由找到的第一个 SPD 入口决定往后的过程。假如对该包的策略是丢弃，则丢弃该包，假如对该包的策略是通过，则 IPSec 过程结束，IP 包就绕过 IPSec 处理，用于网络传输。

（4） 假如对该包的策略是保护，则搜索匹配的安全联盟数据库 SAD（Security Association Database）。如果没有找到 SAD，则唤醒 IKE 用合适的私钥生成 SA 以及 SA 的入口。

（5） 假如找到匹配的 SAD，则进一步的处理就由 SAD 决定。加密、认证或两者都执行，使用传输或隧道模式。然后 IP 包用于网络传输。



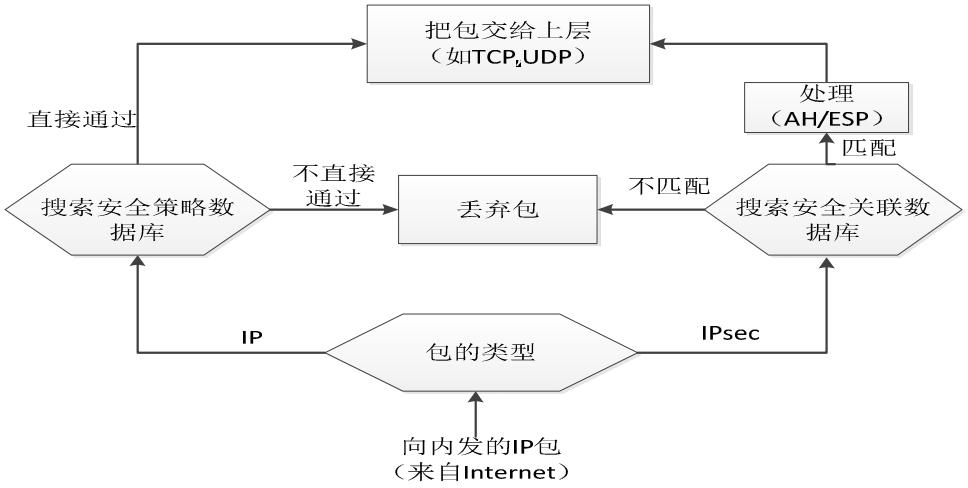
向内发包:

一个到来的 IP 包触发了 IPSec 处理过程，然后执行以下步骤：

（1） IPSec 通过检查 IP 协议（IPv4）或下一个头域（IPv6）来判断该包是一个非安全的 IP 包还是有 ESP 或 AH 头/尾的包。

（2） 假如是非安全的包，IPSec 搜索匹配的 SPD。假如第一个匹配的入口的策略是通过，则处理 IP 头然后将包的正文传递给上一层，如 TCP 层。假如第一个匹配的入口的策略是保护或丢弃，或没有找到匹配的 SPD，则丢弃该包。

（3） 如果是安全的包，IPSec 就搜索 SAD。假如没有找到匹配的 SAD 入口，则丢弃该包；否则 IPSec 执行合适的 ESP 或 AH 过程。然后处理 IP 报头并将包正文发送给上一层，如 TCP 层。

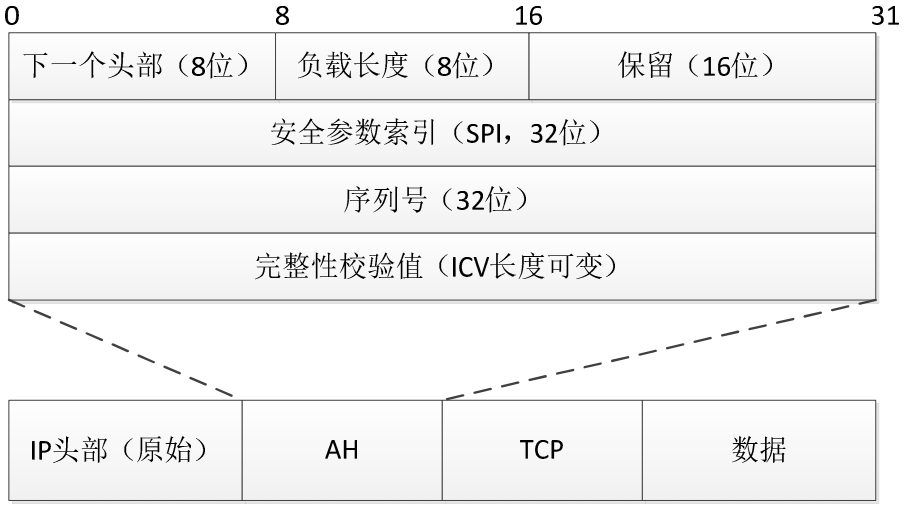


4.安全结构

4.1 安全协议

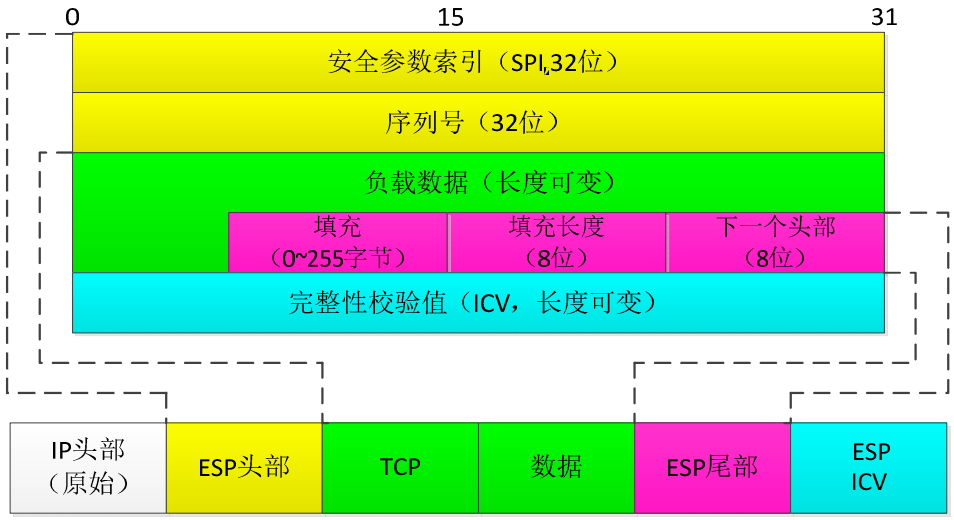
(1). AH（Authentication Header）协议（IP 协议号为 51）提供数据源认证、数据完整性校验和防报文重放功能，它能保护通信免受篡改，但不能防止窃听，适合用于传输非机密数据。AH 的工作原理是在每一个数据包上添加一个身份验证报文头，此报文头插在标准 IP 包头后面，对数据提供完整性保护。可选择的认证算法有 MD5（Message Digest）、SHA-1（Secure Hash Algorithm）等。

如下图，下一个头部（Next Header）表示紧跟在 AH 头部的下一个载荷的类型，即紧跟在 AH 头部数据的协议。负载长度（Payload Length）指出了 AH 的长度，安全参数指引（Security Parameter Index，SPI）字段包含了一个位于接受者端的 SA 标识符。序列号（Sequence Number）字段会随着每一个 SA 数据包的发送而增 1，可用于抵抗重放攻击。完整性校验值 ICV 字段的长度是可变的并且依赖于使用的密码套件。



(2). ESP(Encapsulated Security Payload) 协议（IP 协议号为 50）提供加密、数据源认证、数据完整性校验和防报文重放功能。ESP 的工作原理是在每一个数据包的标准 IP 包头后面添加一个 ESP 报文头，并在数据包后面追加一个 ESP 尾。与 AH 协议不同的是，ESP 将需要保护的用户数据进行加密后再封装到 IP 包中，以保证数据的机密性。常见的加密算法有 DES、3DES、AES 等。同时，作为可选项，用户可以选择 MD5、SHA-1 算法保证报文的完整性和真实性。

如下图，ESP 消息结构中间包含了被加密的负载数据（Payload Data）。安全参数指引与序列号构成了 ESP 头部，而填充（Padding）、填充长度（Padding Length）以及下一个头部（Next Header）字段构成了 ESP 尾部。当需要进行完整性保护时，可以使用一个可选的 ESP ICV 尾部。



在实际进行 IP 通信时，可以根据实际安全需求同时使用这两种协议或选择使用其中的一种。AH 和 ESP 都可以提供认证服务，不过，AH 提供的认证服务要强于 ESP。

4.2 安全联盟（Security Association，SA）

IPsec 在两个端点之间提供安全通信，端点被称为 IPsec 对等体。

SA 是 IPsec 的基础，也是 IPsec 的本质。SA 是通信对等体间对某些要素的约定，例如，使用哪种协议（AH、ESP 还是两者结合使用）、协议的封装模式（传输模式和隧道模式）、加密算法（DES、3DES 和 AES）、特定流中保护数据的共享密钥以及密钥的生存周期等。建立 SA 的方式有手工配置和 IKE 自动协商两种。

SA 是单向的，在两个对等体之间的双向通信，最少需要两个 SA 来分别对两个方向的数据流进行安全保护。同时，如果两个对等体希望同时使用 AH

和 ESP 来进行安全通信，则每个对等体都会针对每一种协议来构建一个独立的 SA。

SA 由一个三元组来唯一标识，这个三元组包括 SPI（Security Parameter Index，安全参数索引）、目的 IP 地址、安全协议号（AH 或 ESP）。

SPI 是用于唯一标识 SA 的一个 32 比特数值，它在 AH 和 ESP 头中传输。在手工配置 SA 时，需要手工指定 SPI 的取值。使用 IKE 协商产生 SA 时，SPI 将随机生成。

通过 IKE 协商建立的 SA 具有生存周期，手工方式建立的 SA 永不老化。IKE 协商建立的 SA 的生存周期有两种定义方式：

基于时间的生存周期，定义了一个 SA 从建立到失效的时间；

基于流量的生存周期，定义了一个 SA 允许处理的最大流量。

生存周期到达指定的时间或指定的流量，SA 就会失效。SA 失效前，IKE 将为 IPsec 协商建立新的 SA，这样，在旧的 SA 失效前新的 SA 就已经准备好。在新的 SA 开始协商而没有协商好之前，继续使用旧的 SA 保护通信。在新的 SA 协商好之后，则立即采用新的 SA 保护通信。

4.3 密钥管理协议

安全联盟密钥管理协议 ISAKMP(Internet Security Association and Key Management Protocol)定义了密钥管理框架。

ISAKMP 由 RFC2408 定义，定义了协商、建立、修改和删除 SA 的过程和包格式。

ISAKMP 是 IKE (Internet Key Exchange)的本质协议，它决定了 IKE 协商包的封装格式，交换过程和模式切换。IKE 是一个混合协议，包括 3 个协议：

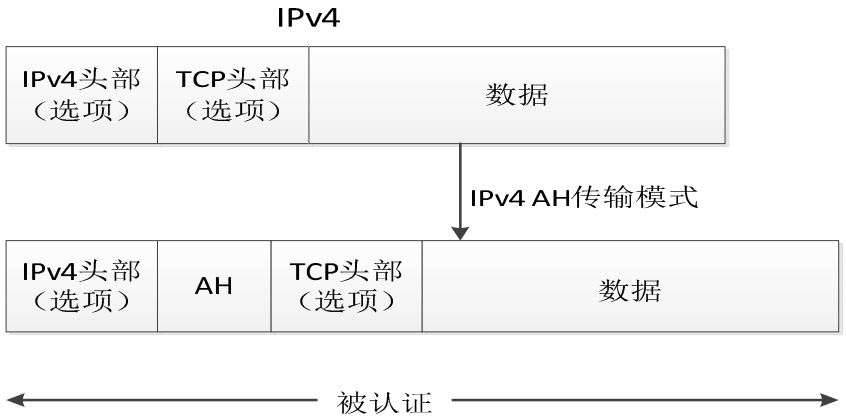
- SKEME（Secure Key Exchange Mechanism）：定义了以认证为目的的公共密钥加密的机制。
- OAKLEY（Key Determination Protocol）：决定了 IPsec 的框架设计，提供在两个 IPSEC 对等体间达成相同加密密钥的基本模式的机制。
- ISAKMP（Internet Security Association and Key Management Protocol）：是 IKE 的核心协议，定义了消息交换的体系结构，包括两个 IPSEC 对等体分组形式和状态转变(定义封装格式和协商包交换的方式)。在配置 IPSEC VPN 的时候，只能设置 ISAKMP，前两个协议不能被设置。

5.封装模式

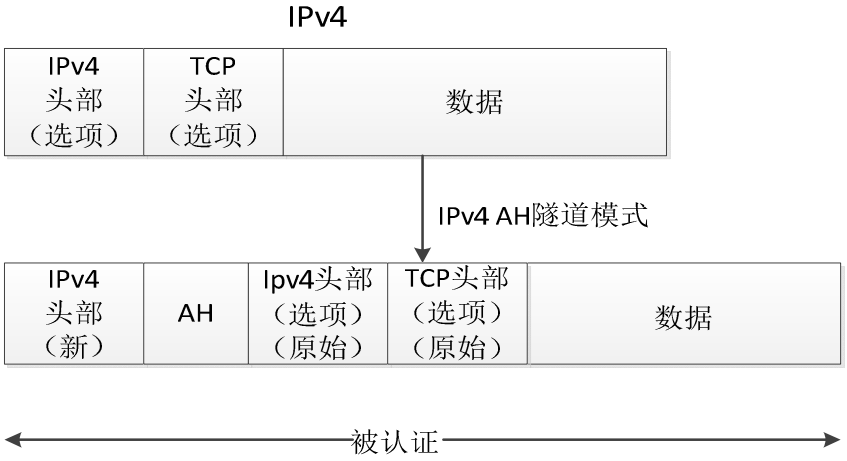
传输（Transport）模式：只是传输层数据被用来计算 AH 或 ESP 头，AH 或 ESP 头以及 ESP 加密的用户数据被放置在原 IP 包头后面。通常，传输模式应用在两台主机之间的通讯，或一台主机和一个安全网关之间的通讯。

隧道（Tunnel）模式：用户的整个 IP 数据包被用来计算 AH 或 ESP 头，AH 或 ESP 头以及 ESP 加密的用户数据被封装在一个新的 IP 数据包中。通常，隧道模式应用在两个安全网关之间的通讯。

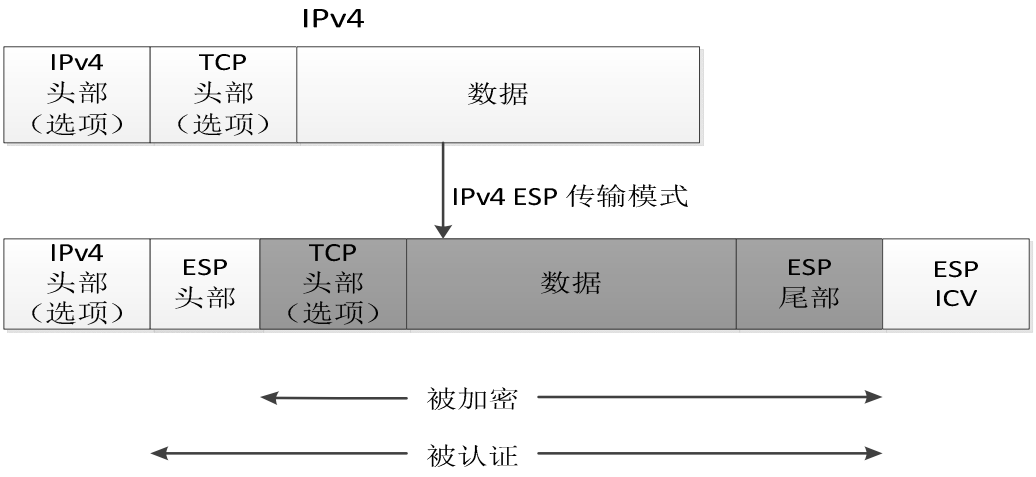
AH 传输模式数据包封装图解（以下以 IPv4 为例）：



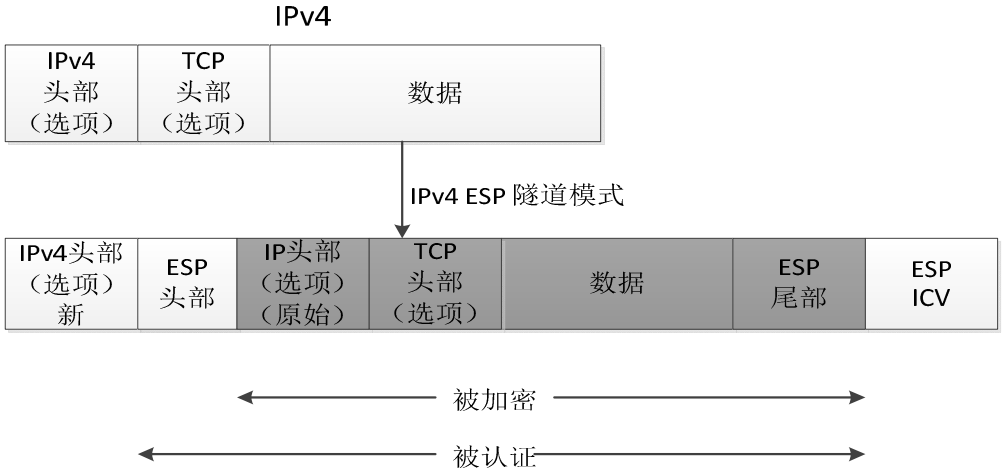
AH 隧道模式数据包封装图解（以下以 IPv4 为例）：



ESP 传输模式（以下以 IPv4 为例）：



ESP 隧道模式（以下以 IPv4 为例）：



6.Internet 密钥交换协议 (Internet Key Exchange，IKE)

6.1 IKE 简介

在实施 IPsec 的过程中，可以使用 IKE 协议来建立 SA，该协议建立在由 ISAKMP（Internet Security Association and Key Management Protocol）定义的框架上。IKE 为 IPsec 提供了自动协商交换密钥、建立 SA 的服务，能够简化 IPsec 的使用和管理，大大简化 IPsec 的配置和维护工作。

6.2 IKE 的交换过程

IKE 使用了两个阶段为 IPsec 进行密钥协商并建立 SA：

(1) 第一阶段，通信各方彼此间建立了一个已通过身份认证和安全保护的通道，即建立一个 ISAKMP SA。第一阶段有主模式（Main Mode）和野蛮模式（Aggressive Mode）两种 IKE 交换方法。

(2) 第二阶段，用在第一阶段建立的安全隧道为 IPsec 协商安全服务，即为 IPsec 协商具体的 SA，建立用于最终的 IP 数据安全传输的 IPsec SA。第二阶段使用的是快速模式（Quick Mode）。

五. VPN

1. VPN 简介：

VPN：虚拟专用网络，在公用网络上建立专用网络，进行加密通讯。主要用于远程访问内网资源，VPN 网关通过对数据包的加密和数据包目标地址的转换实现远程访问。

2. VPN 分类以及相关技术：

VPN 大致上有三种分类，分别如下：

1. VPDN： 远程用户或者移动用户，访问本地网络；
2. VPRN（内部 VPN）：子公司和总部的连接；
3.扩展 VPN：两个联盟公司的连接；

3. VPN 技术原理：

不同的 VPN 技术可以在不同的 OSI 协议层实现如下表所示：

SSL VPN	应用层
Socks5	会话层
IPSec VPN	网络层
PPTP 和 L2TP	数据链路层

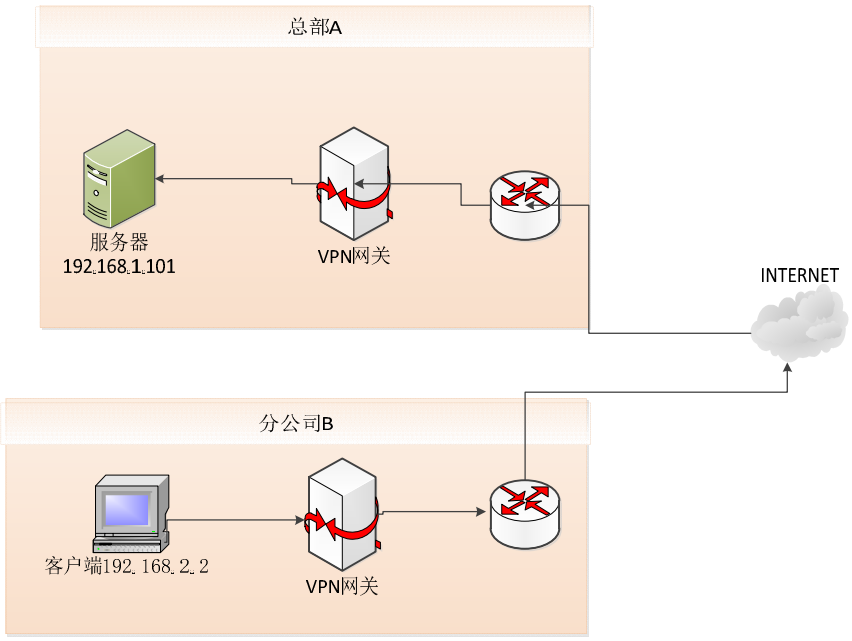
VPN 的工作流程大体相似，其所使用的技术主要体现在传输加密的方面，下面主要对 SSL VPN 和 IPSec VPN 技术做说明：

1. SSL VPN 多用于移动用户访问本地网络，用公钥加密通过 SSL 加密传输的数据，它是建立在应用层上的高层安全协议，无需安装客户端程序；

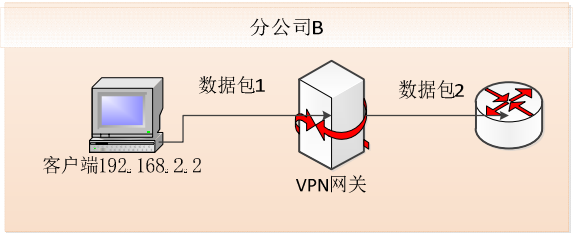
2. IPsec VPN 多用于子公司和总部之间的访问，其使用的是隧道加密方案，隧道将数据包隐藏或者封装在新的数据包内部，从而达到保密的功能，该方式需要安装专门的 IPSec 客户端软件；

4. VPN 工作流程（下文主要对 VPRN 类型进行分析）：

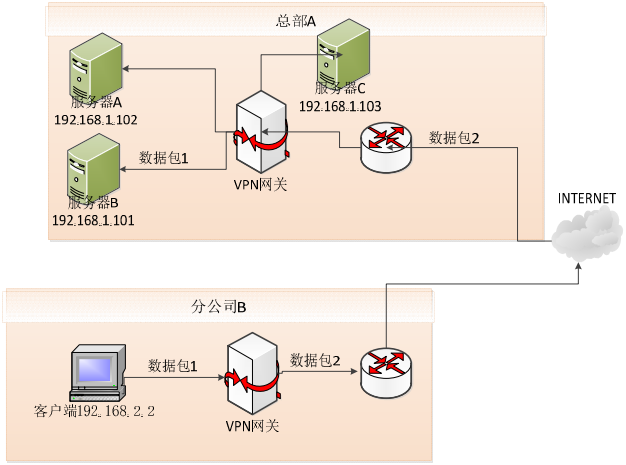
1. VPN 网关采取双网卡结构，内网卡和外网卡，其中外网卡使用公网 IP 接入 Internet；
2. 客户端（192.168.2.2）需要访问总部 A 的服务器（192.168.1.101），其发出的访问数据包的目标地址为服务器的 IP：192.168.1.101，具体流程见下图所示：



3. 分公司 B 的 VPN 网关在接收到客户端（192.168.2.2）发出的访问数据包①时对其目标地址（192.168.1.101）进行检查，发现目标地址属于 A 网络的地址，于是将该数据包①根据所采用的 VPN 技术进行封装，同时 VPN 网关会构造一个新的 VPN 数据包②，并将封装后的原数据包①作为 VPN 数据包②的负载，VPN 数据包的目标地址为 A 网络的 VPN 网关对应的外部地址。



4. 此时分公司 B 的路由器发送数据包 2 到 A 的 VPN 外部地址，A 的 VPN 网关收到数据包 2 后，对数据包进行检查并解包，最终还原成数据包 1 发送给目标地址（192.168.1.101）；



5. 收到消息后，在目标地址（192.168.1.101）看来，消息就是从（192.168.2.2）直接发送过来的一样，而客户端接收消息的过程和发送消息的过程一样，这样就完成了整个 VPN 的通信过程；

另外 VPDN 工作流程和上文 VPRN 类似，区别在于，客户端不需要搭建 VPN 网关设备，并且虚拟传输数据时采用 SSL 协议进行加密；