IPSec 传输模式下 ESP 报文的装包与拆包过程

基础概念部分源自`维基百科`

1. IPSec传输模式

1.1 IPSec 简介

IPSec 全称为：**互联网安全协议**（ Internet Protocol Security，缩写为IPsec），是一个**协议包**，通过对IP协议的分组进行加密和认证来保护IP协议的网络传输协议族（一些相互关联的协议的集合）。

IPSec 定义了在网络层使用的安全服务， 其功能包括数据加密、 对网络单元的访问控制、 数据源地址验证、 数据完整性检查和防止重放攻击。

IPSec 是安全联网的长期方向。 它通过端对端的安全性来提供主动的保护以防止专用网络与 Internet 的攻击。 在通信中， 只有发送方和接收方才是唯一必须了解 IPSec 保护的计算机。 在 `Windows XP 和 Windows Server 2003` 家族中，IPSec 提供了一种能力， 以保护工作组、 局域网计算机、 域客户端和服务器、 分支机构（物理上为远程机构） 、 `Extranet` 以及漫游客户端之间的通信。

1.2 IPSec 的组成

IPsec主要由以下部分组成：

- **认证头（AH）**，为IP数据报提供无连接数据完整性、消息认证以及防重放攻击保护；
  - 认证头分组图示：

0	1	2	3
0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
下一个头	载荷长度	保留	
安全参数索引（SPI）			
串行号			
认证数据（可变长度）			

  - 字段含义：
    - 下一个头：标识被传送数据所属的协议。
    - 载荷长度：认证头包的大小。
    - 保留：为将来的应用保留（目前都置为0）。
    - 安全参数索引：与IP地址一同用来标识安全参数。
    - 串行号：单调递增的数值，用来防止重放攻击。
    - 认证数据：包含了认证当前包所必须的数据。
- **封装安全载荷（ESP）**，提供机密性、数据源认证、无连接完整性、防重放和有限的传输流（traffic-flow）机密性；
- **安全关联（SA）**，提供算法和数据包，提供AH、ESP操作所需的参数。

1.3 IPsec 的设计意图

IPsec被设计用来提供：

- 入口对入口通信安全，在此机制下，分组通信的安全性由单个节点提供给多台机器（甚至可以是整个局域网）；
- 端到端分组通信安全，由作为端点的计算机完成安全操作。

上述的任一模式都可以用来**构建虚拟专用网(**VPN).

1.4 IPsec 与其他互联网协议的对比

IPsec协议工作在OSI模型的第三层，使其在单独使用时适于保护基于 `TCP` 或 `UDP` 的协议（如安全套接子层（ `SSL` ）就不能保护 `UDP` 层的通信流）。这就意味着，与传输层或更高层的协议相比， `IPsec` 协议必须处理可靠性和分片的问题，这同时也增加了它的复杂性和处理开销。相对而言， `SSL/TLS` 依靠更高层的 `TCP` （OSI的第四层）来管理可靠性和分片。

---

2. ESP

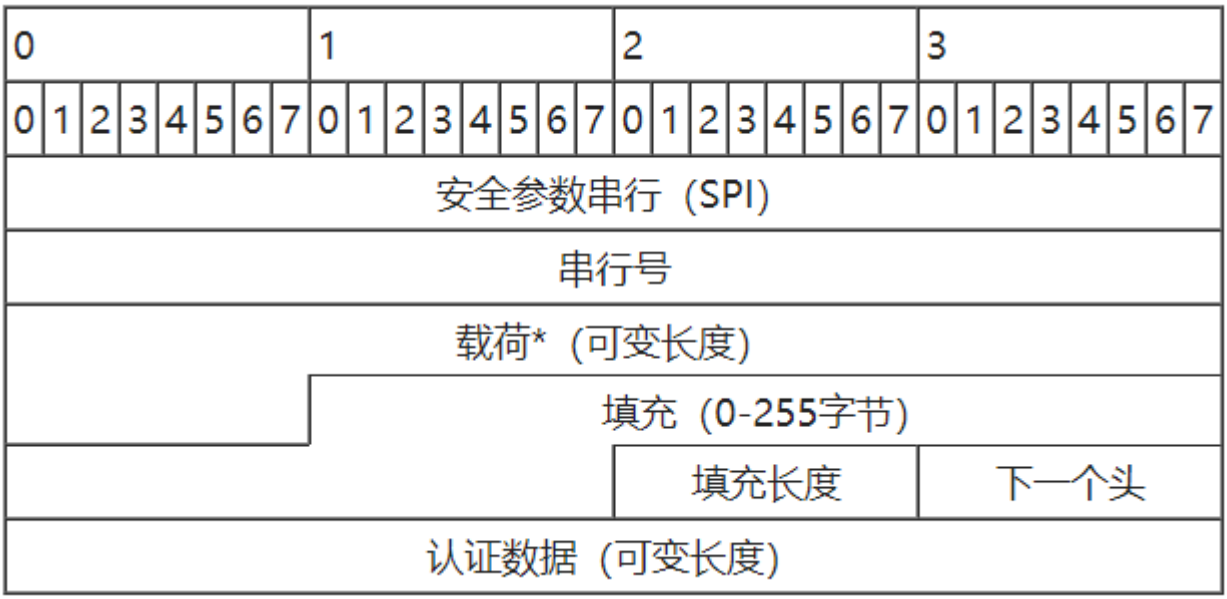
2.1 ESP 简介

ESP 又称 IPsec 封装安全负载(IPsec ESP)是 IPsec 体系结构中的一种主要协议， 其主要设计来在 `IPv4` 和 `IPv6` 中提供安全服务的混合应用。

该协议能够在数据的传输过程中对数据进行完整性度量， 来源认证以及加密， 也可以防止回放攻击。

2.2 ESP 的组成

- ESP 分组图示：



- 字段含义：
  - ESP 头部：
    - 安全参数索引：与IP地址一同用来标识安全参数
    - 序列号：单调递增的数值，用来防止重放攻击。
  - ESP 尾部：

        - 载荷数据：实际要传输的数据。
        - 填充：某些块加密算法用此将数据填充至块的长度。
        - 填充长度：以位为单位的填充数据的长度。
        - 下一个头：标识被传送数据所属的协议。
    - ESP 验证尾部
        - 认证数据：包含了认证当前包所必须的数据。
- 对数据的完整性验证需要计算`SPI`、序列号、载荷数据以及`ESP`尾部。
- 对数据的保密性验证需要计算载荷数据以及ESP尾部。

2.3 ESP 的工作机理

IPsec ESP 通过加密需要保护的数据以及在 `IPsec ESP` 的数据部分放置这些加密的数据来提供机密性和完整性。

ESP 加密采用的是对称密钥加密算法， 能够提供无连接的数据完整性验证、 数据来源验证和抗重放攻击服务。根据用户安全要求， 这个机制既可以用于加密一个传输层的段(如:TCP、 UDP、 ICMP、 IGMP)， 也可以用于加密一整个的 IP数据报。 封装受保护数据是非常必要的， 这样就可以为整个原始数据报提供机密性。 ESP 提供机密性、数据起源验证、 无连接的完整性、 抗重播服务和有限业务流机密性。

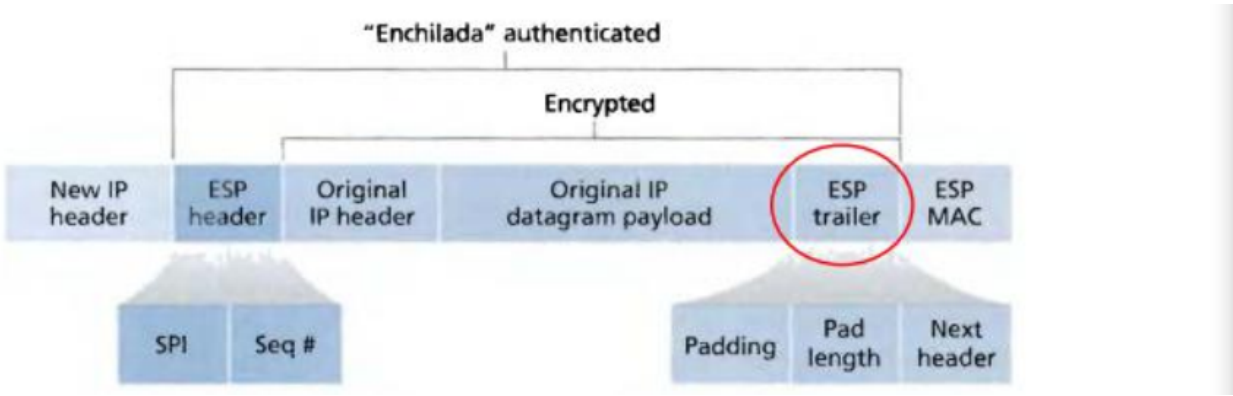
---

3. 传输模式下装包与拆包过程

3.1 传输模式下 ESP 报文结构

3.2 装包过程

1. 在原IP报文末尾添加尾部（`ESP` trailer）信息。尾部包含三部分。由于所选的加密算法可能是块加密，按摩当最后一块长度不够时，需要进行填充（padding），附上填充长度（padding lenght）方便解包时顺利找出用来填充的那一段数据。`Next header`用来表明被加密的数据报文类型。



2. 将原数据报文和刚添加的 `ESP` 尾部信息作为一个整体进行加密， 具体的加密算法由`密钥`和 `SA` 给出。
3. 在第 2 步得到的加密数据前添加 `ESP Header`。 `ESP Header` 由 `SPI` 和 序号(Sequence number)两部分组成。 加密数据与 `ESP` 头合称为 `"enchilada"`。
4. 附加`完整性度量结果`(ICV, Integrity check value)。 对第三步得到的`"enchilada"`做摘要， 得到一个完整性度量值， 并附在 `ESP` 报文的尾部。
5. 将原 `IP` 头放回到第 4 步后形成的报文的头部前， 组织成一个新的 `IP` 报文。

3.3 拆包过程

接收端在收到一个 `ESP` 包之后，若不对这个包进行处理，就无法得知它究竟处于通道模式，还是传送模式。根据对这个包进行处理的 `SA`，便可知道它到底处在什么模式下。所以，我们的拆包过程可如下操作：

1. 接收方收到报文之后，发现协议类型是 50， 知道这是一个 `IPSec` 包。 首先查看 `ESP` 头， 通过里面的 `SPI` 决定数据报文所对应的 `SA`。
2. 计算 `"enchilada"` 部分的摘要， 与附在末尾的 `ICV` 做对比， 如果一样， 说明数据完整； 否则断定收到的报文已经不是原来的报文了。
3. 检查 `Seq` 里的顺序号， 保证数据是"新鲜"的， 不是回放攻击。
4. 根据 `SA` 所提供的加密算法和密钥， 解密被加密过的数据 `"enchilada"`。 得到原 `IP` 报文的数据部分和 `ESP` 尾部(trailer)。
5. 根据 `ESP` 尾部的填充长度信息， 可以找出填充字段的长度， 删去后就得到原来的 `IP` 报文。
6. 根据获取的原 `IP` 包目标地址进行转发。