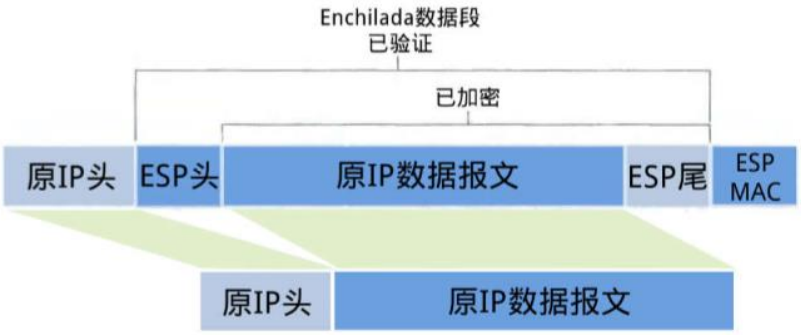

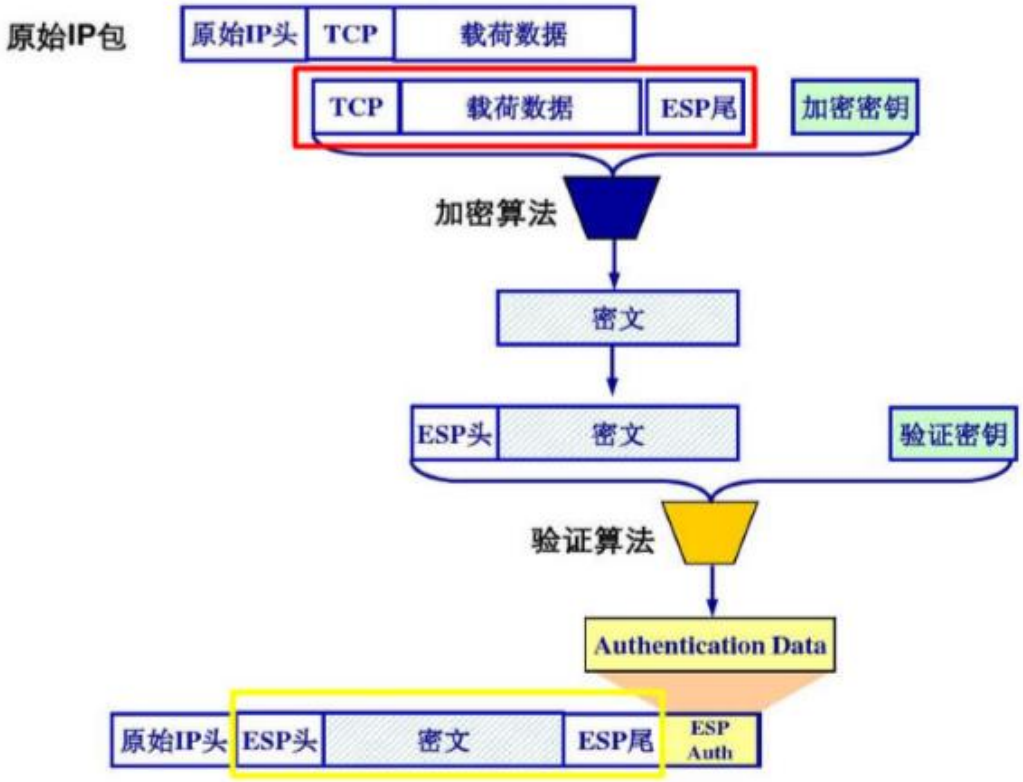

【装包过程】

1. 在 IP 数据报文的末尾添加 ESP 尾部信息。尾部包含三部分，分别是 padding、padding length、next=ip
   a) Padding：当加密算是块加密时，数据的长度需要是每块长度的整数倍，此时当数据长度不够整数倍时，需要进行填充，直至可以分成若干完整的块。
   b) Padding length：是上述 padding 的长度记录，方便在后期的拆包过程中找到非原始数据
   c) Next=ip：当前报文的类型，如 next=tcp

2. 对 IP 数据报文与 ESP 尾部使用加密密钥以及加密算法（SA 安全关联）进行加密
3. 在原 IP 头和第 2 步得到的结果中间添加 ESP 头。ESP 头包含两部分，分别是:SPI 和序列号。
    a) SPI：安全参数索引，使用这个数值判断对应的安全关联
    b) 序列号：通过序列号判断当前的包的状态，检查回放攻击
4. 对 ESP 头和第 2 步得到的结果使用验证密钥和验证算法生成 authentication data，添加到最后方作为 ESP MAC
5. 装包过程完成，最后的结构为：原始 IP 头->第 3 步生成的 ESP 头->第 2 步得到的加密结果->第 4 步得到的 ESP MAC

【拆包过程:】

（对装包的部分进行反向验证）

1. 对 ESP 头、密文、ESP 尾三个部分计算摘要，并与 ESP MAC 部分进行对比，如果对比结果不同，就证明与装包过程时进行的验证结果是不同的，也就是说数据被篡改了
2. 检查 ESP 头的两个组成部分
    a) SPI：利用这个数值判断对应的安全关联 SA
    b) 序列号：检查是否是回放攻击。
3. 根据第 2 步中的 SPI 对应的 SA 得到加密算法与密钥，同时完成解密，得到在装包过程中第 2 步操作的 IP 数据报文与 ESP 尾部
4. 根据 ESP 尾部中的 padding length 得到后续添加的 padding 部分的长度，并将这部分截去，同时将 next=iP 中的 IP 类型记为真正的 IP 类型后，得到的最终结果就是拆包后的结果。