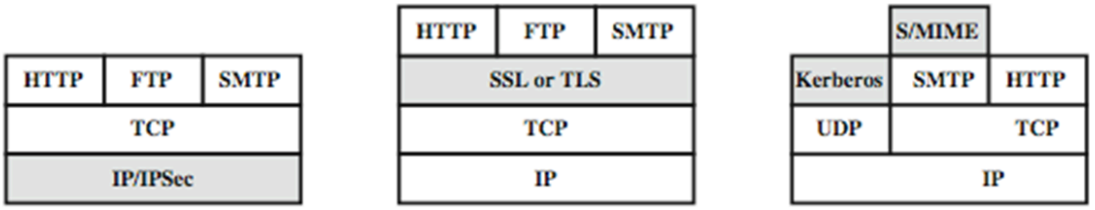
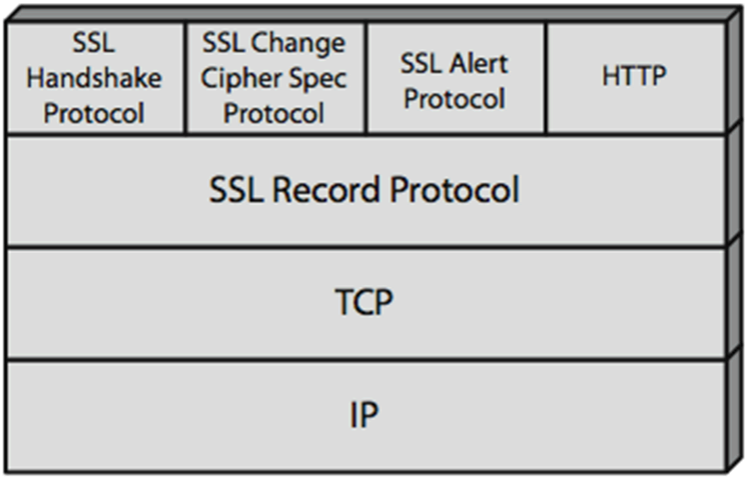
传输层安全

- 面临的威胁：完整性、机密性、拒绝服务和认证
- Web流量安全方法：



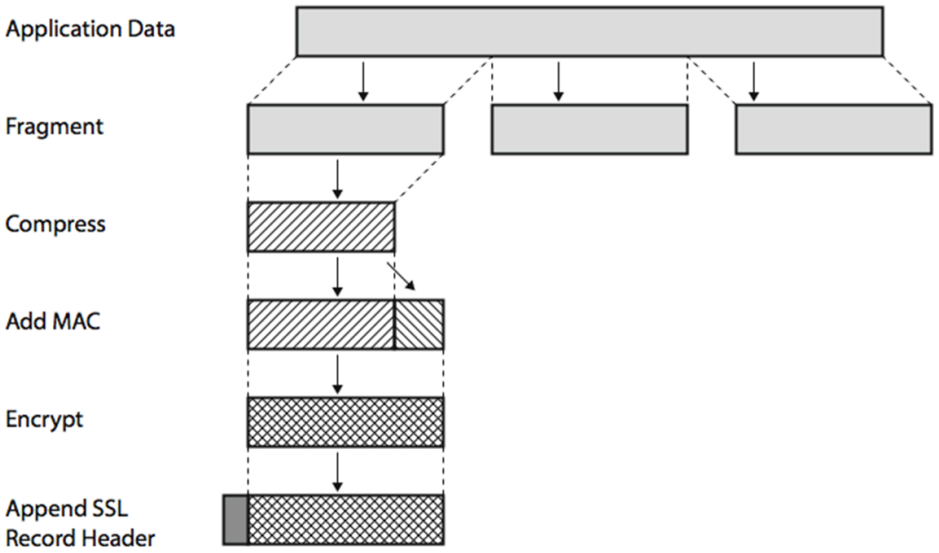
(a) Network Level (b) Transport Level (c) Application Level

- SSL（安全套接层）
  - 采用TCP提供一种可靠的端到端的安全服务（提供传输层安全服务）
  - 由两层协议组成
  - 体系结构

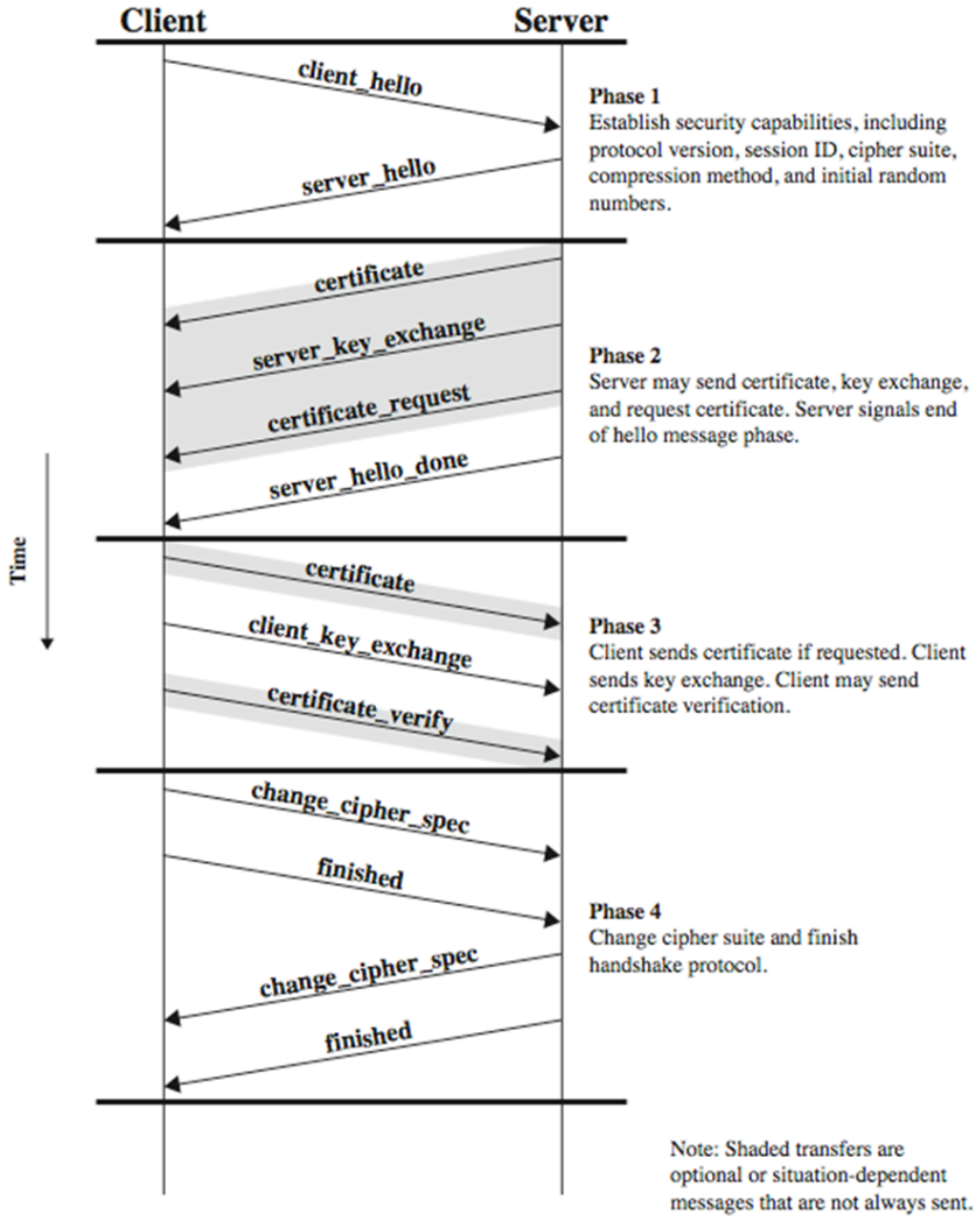


    - SSL连接
      - 一个短暂的，点对点的，传输连接
      - 每一条连接与一个会话相关联
    - SSL会话
      - 客户与服务器之间的一种关联
      - 通过握手协议来创建
      - 定义了密码安全参数集合
      - 多个安全连接之间共享
  - 记录协议服务
    - 机密性
      - 握手协议定义了一个可以用于加密SSL载荷的传统加密共享密钥
      - AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
      - 信息在加密之前被压缩
    - 消息完整性

        - 一个用于产生消息认证码（MAC）的共享密钥
        - 与HMAC相似，但是有不同的填充方法
    - 记录协议操作



    - 修改密码规格协议
    - 警报协议
    - 握手协议

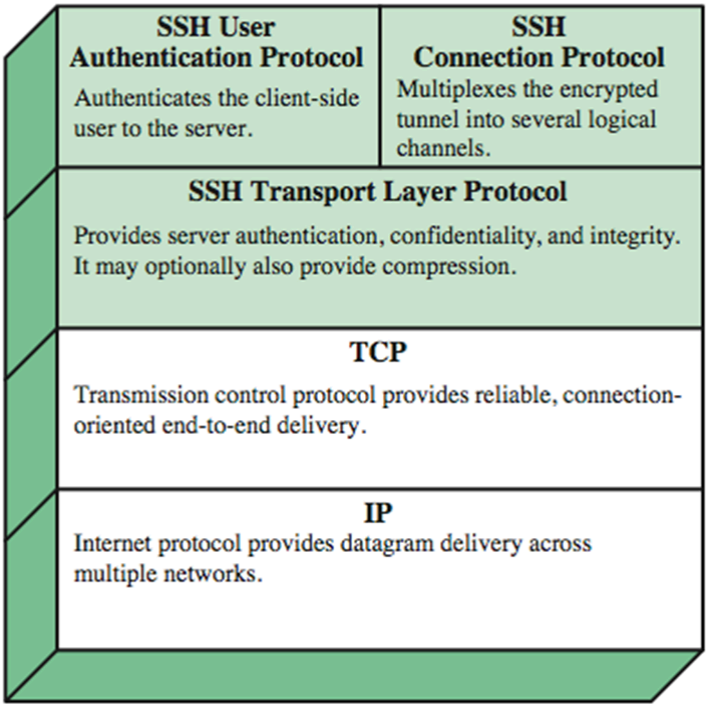


- HTTPS
  - 用 HTTP 和 SSL/TLS 的结合来实现网络浏览器和服务器之前的安全通信
    - 规范文档可参阅 RFC2818
    - 在SSL 或 TLS之上的HTTP 没有根本性区别
  - 使用 https:// URL 而不是 http://
    - 使用端口 443， 而不是 80
  - 加密
    - URL, 文件内容, 表单内容, cookies, HTTP 报头的内容

HTTPS 使用

- 连接初始化
  - TLS 握手，然后 HTTP 请求
- 连接关闭
  - 在 HTTP 记录有 “Connection: close”
  - TLS 层 交换 close_notify 警告
  - 接着关闭 TCP 连接
  - TLS实例在关闭连接之前，发起一个关闭警报的交换

- SSH
  - 协议栈



    - SSH传输层协议
      - 基于一对服务器/主机密钥，服务器认证发生在传输层
        - 服务器认证需要客户端提前知道服务器的公共主机密钥
      - 分组交换
        - 建立 TCP 连接
        - 接着交换消息
          - 身份识别字符串交换, 算法协商, 密钥交换,密钥交换结束, 服务请求
        - 使用特定的分组格式
    - SSH用户认证协议
      - 提供用户向服务器证明自己身份的方法

- 三种消息类型:

  SSH_MSG_USERAUTH_REQUEST

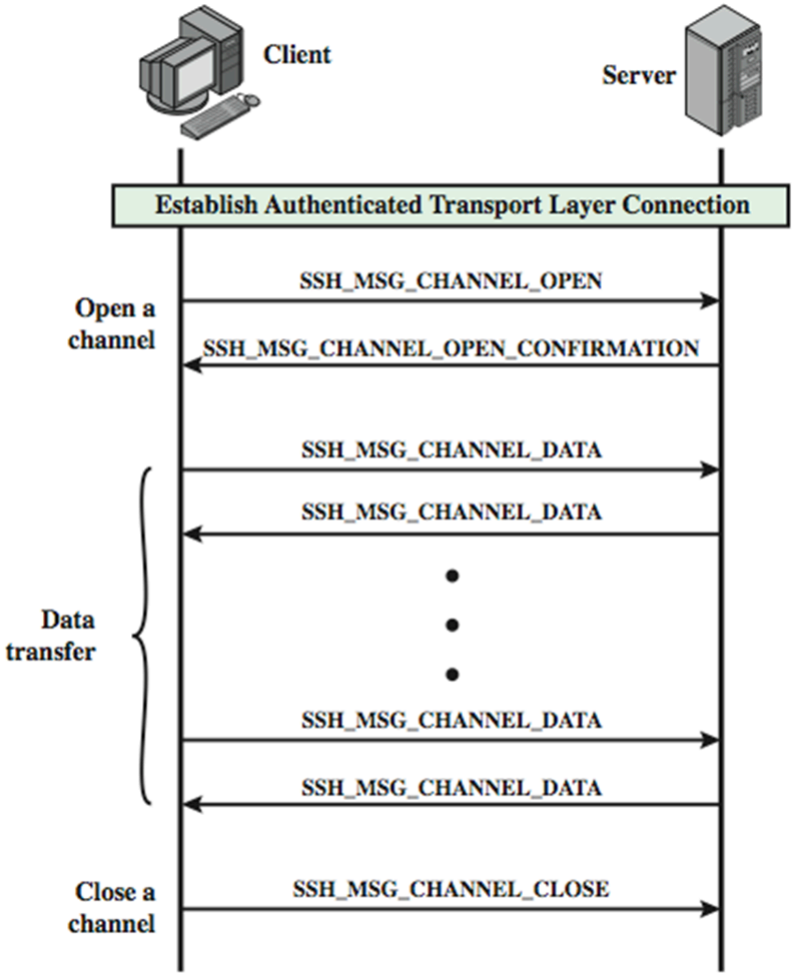
  SSH_MSG_USERAUTH_FAILURE

  SSH_MSG_USERAUTH_SUCCESS

- 使用认证方法

  公开密钥, 口令密钥, 基于主机

- SSH连接协议
  - 在SSH 传输层协议运行
  - 假设使用了安全的认证连接
  - 用一个通道虚拟出多条逻辑信道
- SSH连接协议交换



- 端口转发
  - 将任何不安全的TCP连接转换成安全的SSH连接
    - SSH 传输层协议在SSH客户端和服务器 建立一个TCP连接
    - 客户端流量重定向到当地SSH，通过隧道转发，接着远程SSH传送到服务器
  - 支持两种类型的端口转发
    - 本地转发
    - 远程转发