- HTTPS
  - HTTPS解决了HTTP的3个问题
    - 通信内容暴露。
      - 因为HTTP报文使用明文传输。
    - 无法确认通信方。
      - 因为任何人都可以对服务器发送请求，所以服务器无法确定哪些请求是恶意请求。
        - 可信的第三方机构发布的证书可以证明服务器端或客户端的身份。
    - 通信内容被篡改。
      - 可能遭遇中间人攻击，导致通信内容被篡改。
  - HTTPS是身披SSL的HTTP
    - HTTPS并不是应用层的新协议，只是HTTP与TCP通信的接口加了SSL协议，在HTTPS过程中，HTTP先和SSL通信，SSL再和TCP通信。
      - 所有运行在应用层的协议（SMTP，Telnet）都可以使用SSL协议。
  - 加密技术
    - 近代加密算法都是公开的，通过密钥对内容进行加密。
    - SSL采用的2种加密技术
      - 对称加密
        - 加密和解密都使用同一套密钥。
          - 对称加密的缺陷在于传输密文之后也需要把密钥传输给对方，对方才能进行解密，然而在传输密钥的这个过程中，密钥可能被窃取。
      - 非对称加密
        - 非对称加密存在私钥和公钥，私钥需要保密，公钥可以公开。
          - 密文发送方使用对方的公钥进行加密，密文接收方在收到密文之后，使用自己的私钥进行解密。
            - 非对称加密的2个缺陷
              - 加密消耗的资源过多。
                - 解决方案是采用非对称加密和对称加密结合的方式。
              - 不能确定公钥是否在传输过程中被攻击者替换。
                - 采用数字证书认证机构（CA）颁发的公钥证书。（公钥证书也可以被称为数字证书或证书）。
                  - 数字证书认证的流程
                    - 首先，服务器运营人员去向CA申请公钥。
                    - CA确认申请者的身份之后，会为申请者分配公钥，并对这个公钥做数字签名（数字签名就是利用CA的私钥进行加密），并把签名后的公钥放进数字证书发送给申请者。
                    - 服务器端在通信的时候会把这个数字证书发送给客户端。
                    - 客户端在接收到数字证书之后，会使用CA的公钥对数字签名进行验证，一旦认证通过，客户端就可以知道服务器端的公钥是没有被篡改的。
                  - 数字证书认证过程必须保证CA的公钥可以安全的转交给客户端，这一般用直接放进浏览器内部进行实现的。
    - HTTPS采用对称加密和非对称加密并用的方法。
      - 在交换密钥阶段使用非对称加密，在交换报文阶段使用对称加密 。
        - 非对称加密阶段
          - 使用非对称加密对报文交换阶段需要使用的对称加密的密钥进行加密传输。
        - 对称加密阶段
          - 使用传输好的密钥进行对称加密。

XMind:ZEN | 试用模式