本讲主题

安全电子邮件基本原理

电子邮件安全威胁

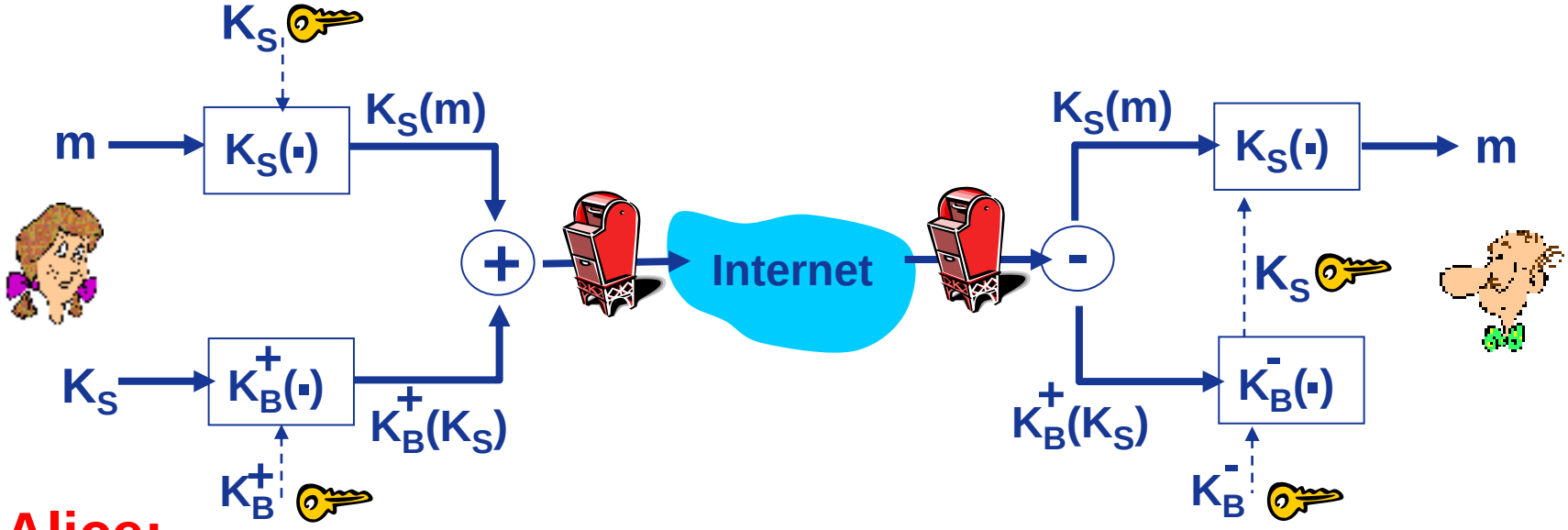
- 垃圾邮件
  - 增加网络负荷，占用服务器空间
- 诈骗邮件
  - 能迅速让大量受害者上当
- 邮件炸弹
  - 短时间内向同一邮箱发送大量电子邮件
- 通过电子邮件/附件传播网络蠕虫/病毒
- 电子邮件欺骗、钓鱼式攻击

电子邮件安全需求

- 机密性
  - 只有真正的接收方才能阅读邮件
- 完整性
  - 电子邮件在传输过程中不被修改
- 身份认证性
  - 电子邮件的发送者不被假冒
- 抗抵赖性
  - 发信人无法否认发过电子邮件

安全电子邮件基本原理

- 邮件具有单向性和非实时性
  - 不能通过建立隧道来保证安全，只能对邮件本身加密
- Alice期望向Bob发送机密邮件m



Alice:

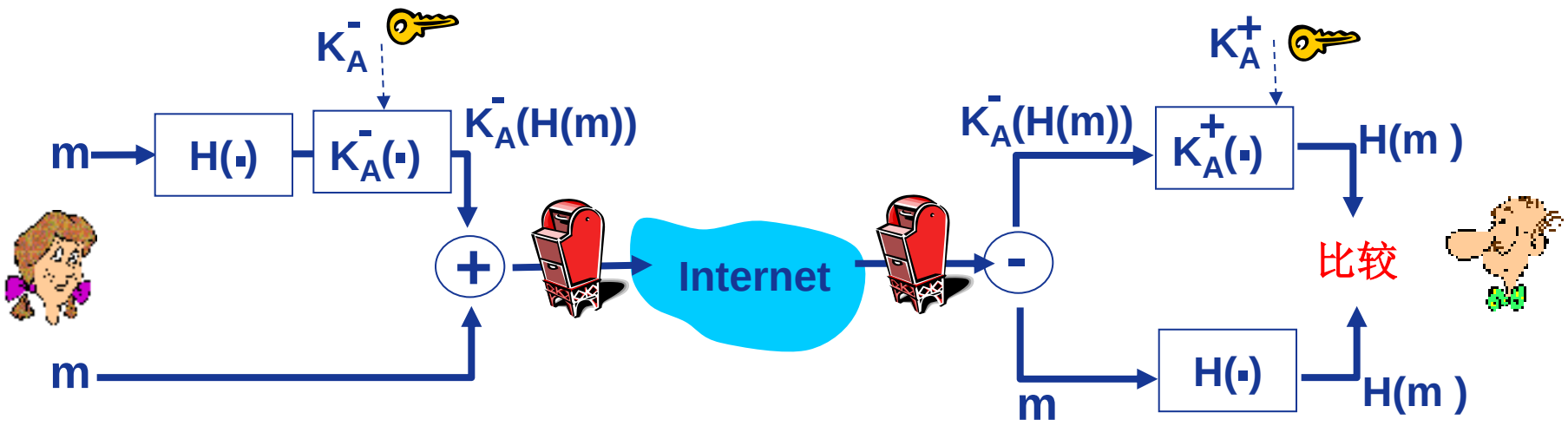
- 生成随机对称密钥, K_S
- 利用K_S加密报文 (为了效率)
- 同时，利用Bob的公钥加密K_S
- 将$K_S(m)$和$K_B^+(K_S)$发送给Bob

Bob:

- 利用他的私钥解密$K_B^+(K_S)$，获得K_S
- 利用K_S解密$K_S(m)$恢复m

安全电子邮件基本原理

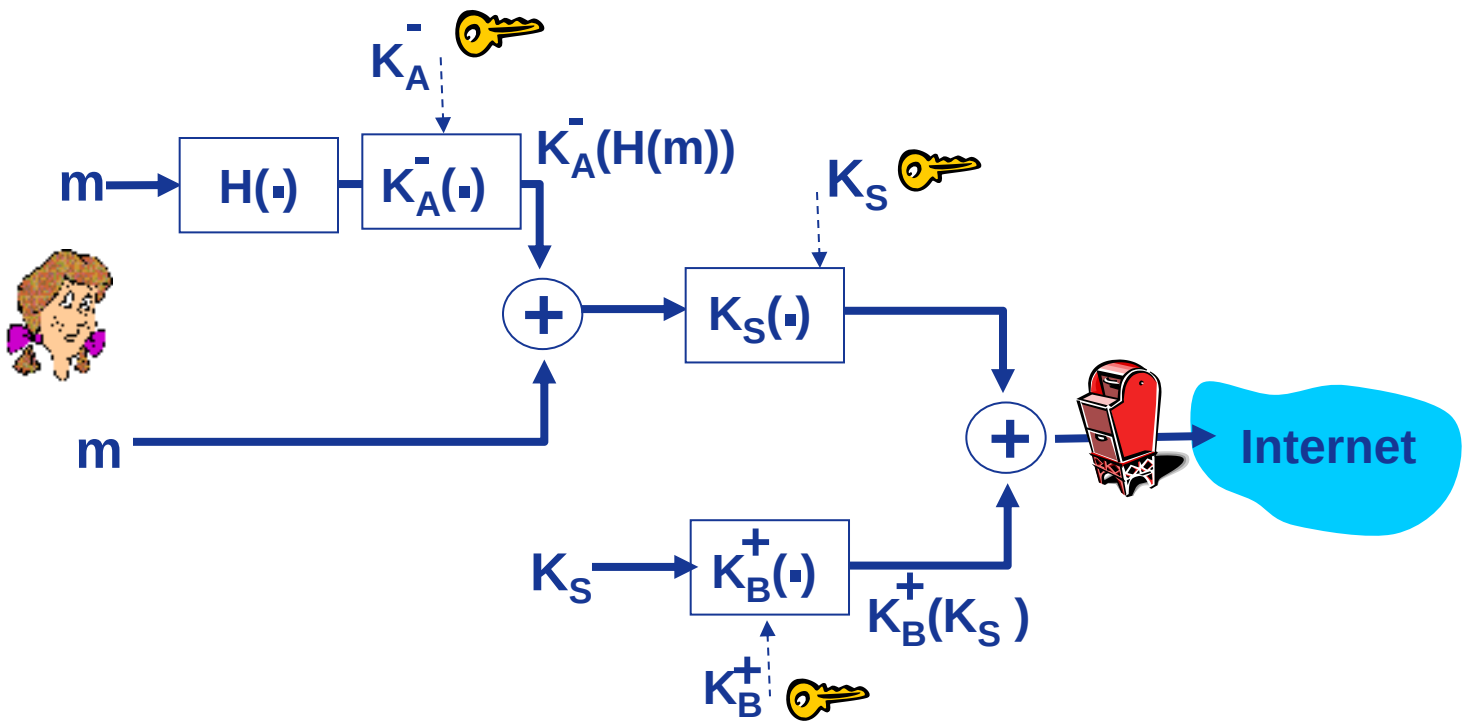
- Alice期望提供发送者认证与报文完整性



- Alice对报文进行数字签名
- 发送报文（明文）和数字签名

安全电子邮件基本原理

- Alice期望提供保密、发送者认证与报文完整性



Alice使用3个密钥: 她自己的私钥、Bob的公钥和新生成的对称密钥

本讲主题

安全电子邮件标准

PGP标准

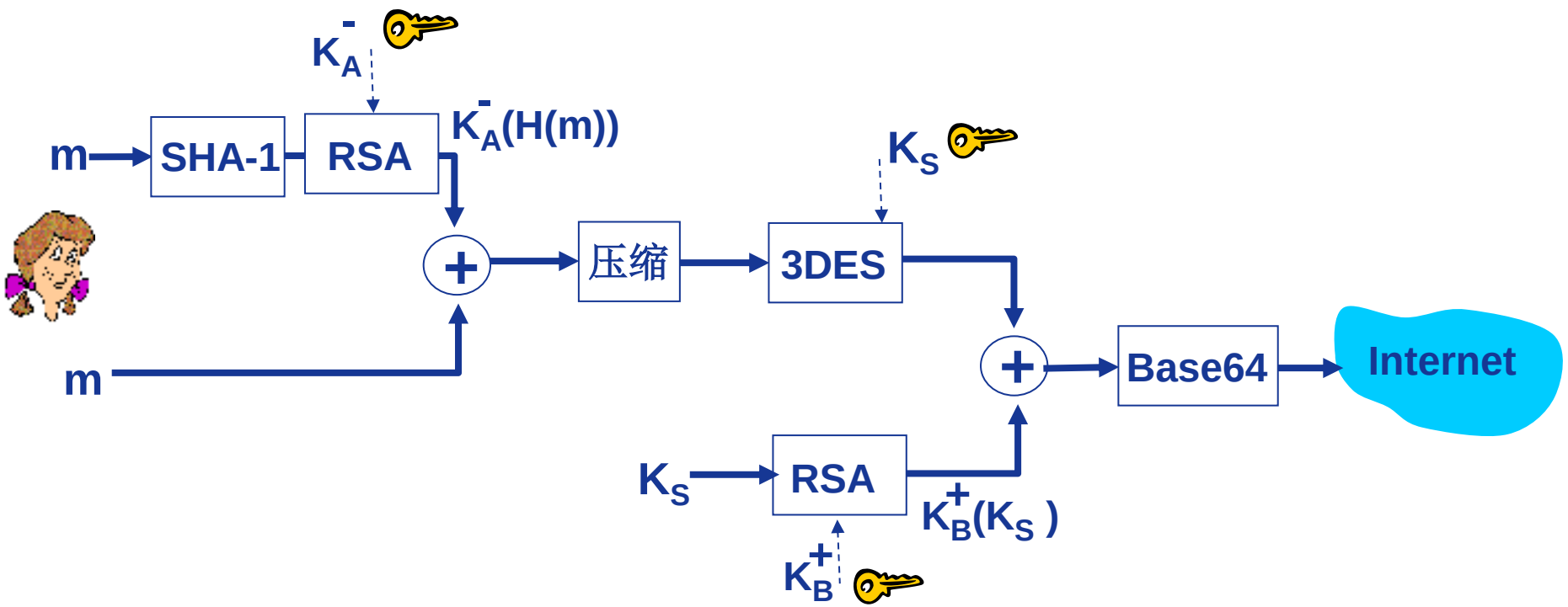
- PGP（Pretty Good Privacy）标准
  - Philip Zimmermann于1991年发布PGP 1.0
    - 事实上标准
  - 可在各种平台（Windows、UNIX等）免费运行
  - 还可用于普通文件加密及军事目的
  - 所用算法被证实为非常安全：
    - 公钥加密算法：RSA、DSS或Diffie-Hellman
    - 对称加密算法：CAST、3DES或IDEA
    - 散列算法：MD5或SHA-1

PGP标准

- PGP特点：
  - 对邮件内容进行数字签名，保证信件内容不被篡改
  - 使用公钥和对称加密保证邮件内容机密且不可否认
  - 公钥的权威性由收发双方或所信任的第三方签名认证
  - 事先不需要任何保密信道来传递对称的会话密钥

PGP功能框架

- Alice期望PGP提供保密、发送者认证与报文完整性



PGP密钥

- 安装PGP时，软件为用户生成一个公开密钥对
  - 公钥放置用户网站或某公钥服务器上
  - 私钥则使用用户口令进行保护
    - 用户为随机生成的RSA私钥指定一个口令，只有给出口令才能将私钥释放出来使用
- PGP公钥认证机制与传统CA差异较大：
  - PGP公钥可以通过可信的Web认证
  - 用户可以自己认证任何其信任的“公钥/用户名”对
  - 用户还可以为其他公钥认证提供“担保”
- 防止篡改公钥的方法（Alice）：
  - 直接从Bob手中得到其公钥
  - 通过电话认证密钥
  - 从双方信任的David那里获得Bob的公钥
  - 通过CA

本讲主题

Web应用安全

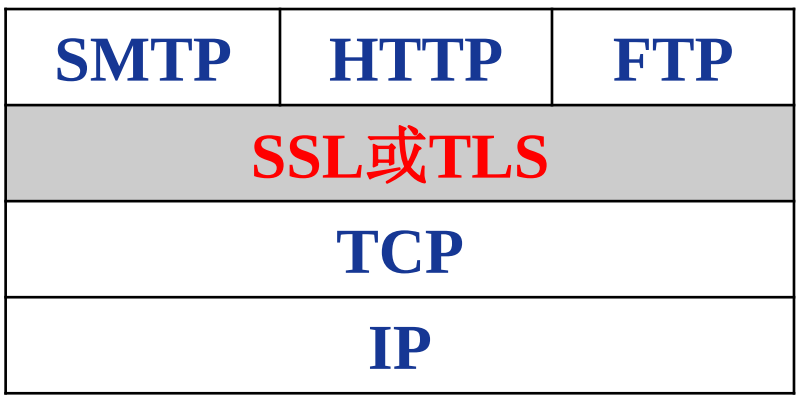
基于应用层实现Web安全

- 为特定应用定制特定安全服务，将安全服务直接嵌入在应用程序中

<table>
<tr><td rowspan="3">Kerberos</td><td>S/MIME</td><td>PGP</td><td>SET</td><td></td></tr>
<tr><td colspan="2">SMTP</td><td>HTTP</td><td>FTP</td></tr>
<tr><td colspan="4">SSH</td></tr>
<tr><td>UDP</td><td colspan="4">TCP</td></tr>
<tr><td colspan="5">IP</td></tr>
</table>

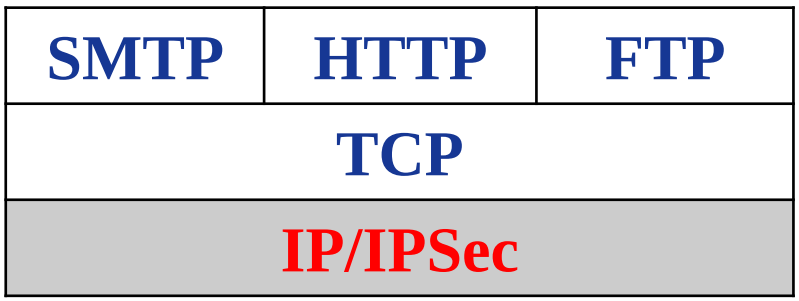
基于传输层实现Web安全

- SSL或TLS可作为基础协议栈的组成部分，对应用透明
  - 也可直接嵌入到浏览器中使用
- 使用SSL或TLS后，传送的应用层数据会被加密
  - 保证通信的安全



基于网络层实现Web安全

- IPSec提供端到端（主机到主机）的安全机制
  - 通用解决方案
- 各种应用程序均可利用IPSec提供的安全机制
  - 减少了安全漏洞的产生



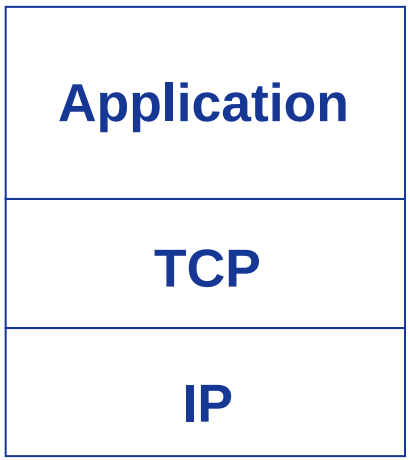
本讲主题

安全套接字层（SSL）（1）

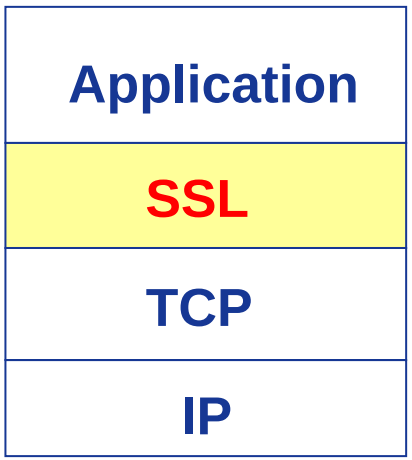
SSL: Secure Sockets Layer

- 广泛部署的安全协议
  - 几乎所有浏览器和Web服务器都支持
  - https
- 实现：Netscape
- 变体：TLS(RFC 2246)
- 提供：
  - 机密性(confidentiality)
  - 完整性(integrity)
  - 认证(authentication)
- 最初目标:
  - Web电子商务交易
  - 加密(尤其信用卡号)
  - Web服务器认证
  - 可选的客户认证
  - 方便与新商户的商务活动(minimum hassle)
- 可用于所有基于TCP的网络应用
  - 安全socket接口

SSL和TCP/IP



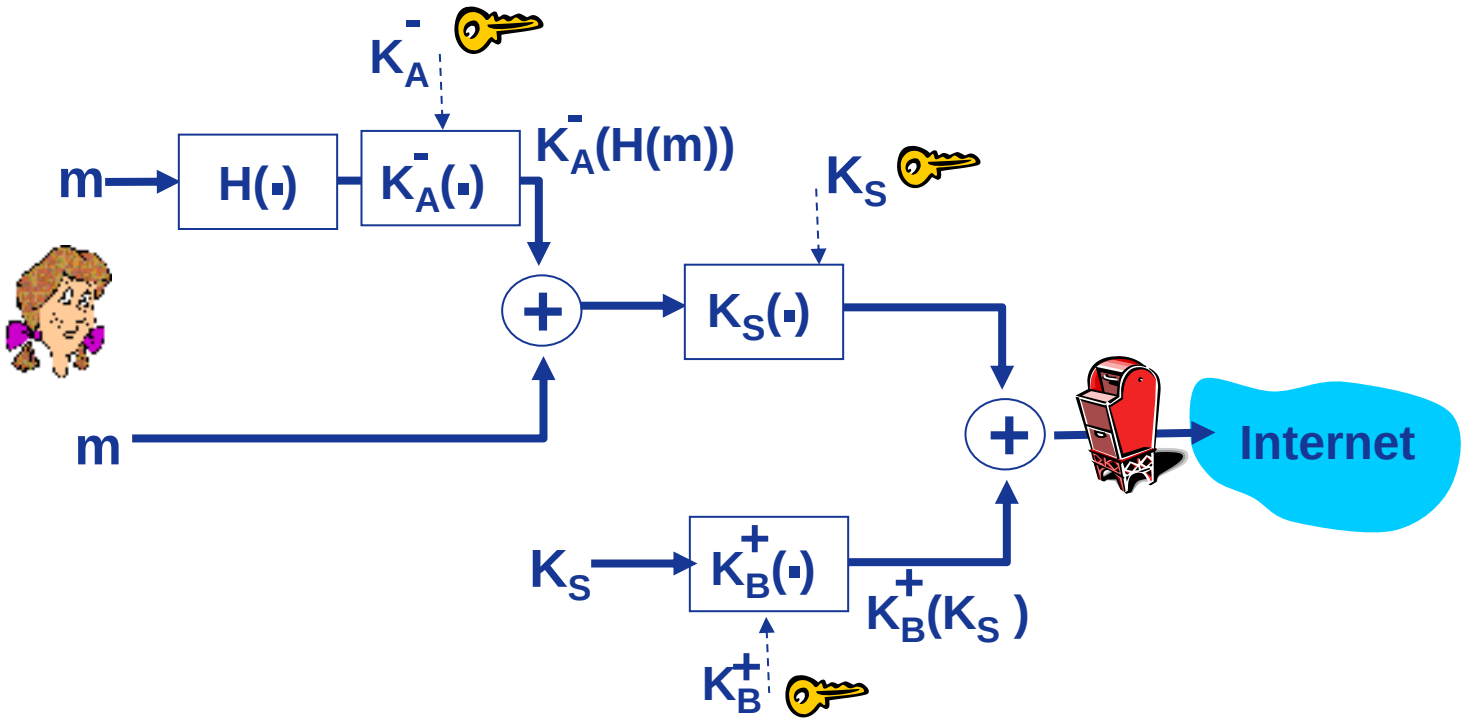
正常应用



采用SSL的应用

- SSL为网络应用提供应用编程接口 (API)
- C语言和Java语言的 SSL库/类可用

可以像PGP那样实现某些安全功能

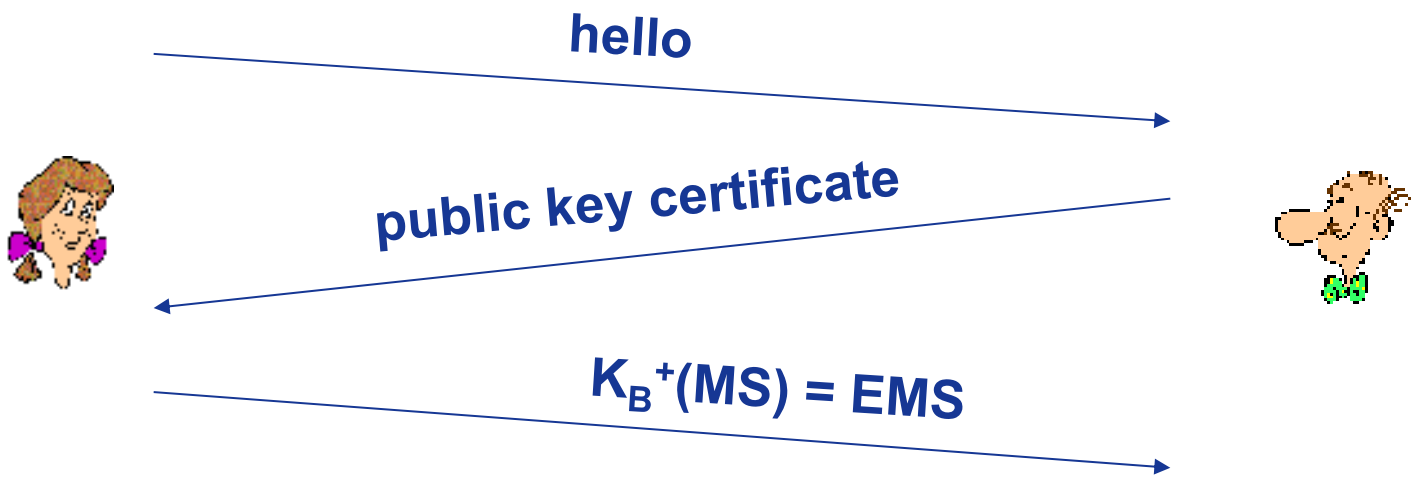


- 但是，需要发送字节流以及交互数据
- 需要一组密钥用于整个连接
- 需要证书交换作为协议的一部分：握手阶段

简化的(Toy)SSL: 一个简单的安全信道

- 握手(handshake): Alice和Bob利用他们的证书、私钥认证（鉴别）彼此，以及交换共享密钥
- 密钥派生(key derivation): Alice和Bob利用共享密钥派生出一组密钥
- 数据传输(data transfer): 待传输数据分割成一系列记录
- 连接关闭(connection closure): 通过发送特殊消息，安全关闭连接

简化的SSL：一个简单的握手过程



MS: 主密钥

EMS: 加密的主密钥

简化的SSL：密钥派生

- 不同加密操作使用不同密钥会更加安全
  - 例如：报文认证码(MAC)密钥和数据加密密钥
- 4个密钥：
  - K_c = 用于加密客户向服务器发送数据的密钥
  - M_c = 用于客户向服务器发送数据的MAC密钥
  - K_s = 用于加密服务器向客户发送数据的密钥
  - M_s =用于服务器向客户发送数据的MAC密钥
- 通过密钥派生函数(KDF)实现密钥派生
  - 提取主密钥和（可能的）一些额外的随机数，生成密钥

简化的SSL：数据记录

- 为什么不直接加密发送给TCP的字节流?
  - MAC放到哪儿?
    - 如果放到最后，则只有全部数据收全才能进行完整性认证。
  - e.g., 对于即时消息应用， 在显示一段消息之前，如何针对发送的所有字节进行完整性检验?
- 方案：将字节流分割为一系列记录
  - 每个记录携带一个MAC
  - 接收方可以对每个记录进行完整性检验
- 问题：对于每个记录， 接收方需要从数据中识别出MAC
  - 需要采用变长记录

简化的SSL：序列号

- 问题: 攻击者可以捕获和重放记录或者重新排序记录
- 解决方案: 在MAC中增加序列号
  - MAC = MAC(M_x, sequence||data)
  - 注意: 记录中没有序列号域

- 问题: 攻击者可以重放所有记录
- 解决方案: 使用一次性随机数(nonce)

简化的SSL：控制信息

- 问题: 截断攻击
  - 攻击者伪造TCP连接的断连段，恶意断开连接
  - 一方或双方认为对方已没有数据发送
- 解决方案: 记录类型, 利用一个类型的记录专门用于断连
  - type 0用于数据记录；type 1用于断连
- MAC = MAC(M_x, sequence||type||data)

简化的SSL：总结

bob.com

hello

certificate, nonce

$K_B^+(MS) = EMS$

type 0, seq 1, data

type 0, seq 2, data

type 0, seq 1, data

type 0, seq 3, data

type 1, seq 4, close

type 1, seq 2, close

加密的

本讲主题

安全套接字层（SSL）（2）

简化的SSL不完整

- 每个域多长？
- 采用哪种加密协议？
- 需要协商吗？
  - 允许客户与服务器支持不同加密算法
  - 允许客户与服务器在数据传输之前共同选择特定的算法

SSL握手过程（1）

1. 客户发送其支持的算法列表，以及客户一次随机数(nonce)
2. 服务器从算法列表中选择算法，并发回给客户：选择 + 证书 + 服务器一次随机数
3. 客户验证证书，提取服务器公钥，生成预主密钥(pre_master_secret)，并利用服务器的公钥加密预主密钥，发送给服务器
4. 客户与服务器基于预主密钥和一次随机数分别独立计算加密密钥和MAC密钥
5. 客户发送一个针对所有握手消息的MAC
6. 服务器发送一个针对所有握手消息的MAC

SSL握手过程(2)

最后2步的意义：保护握手过程免遭篡改

- 客户提供的算法，安全性有强、有弱
  - 明文传输
- 中间人攻击可以从列表中删除安全性强的算法
- 最后2步可以预防这种情况发生
  - 最后两步传输的消息是加密的

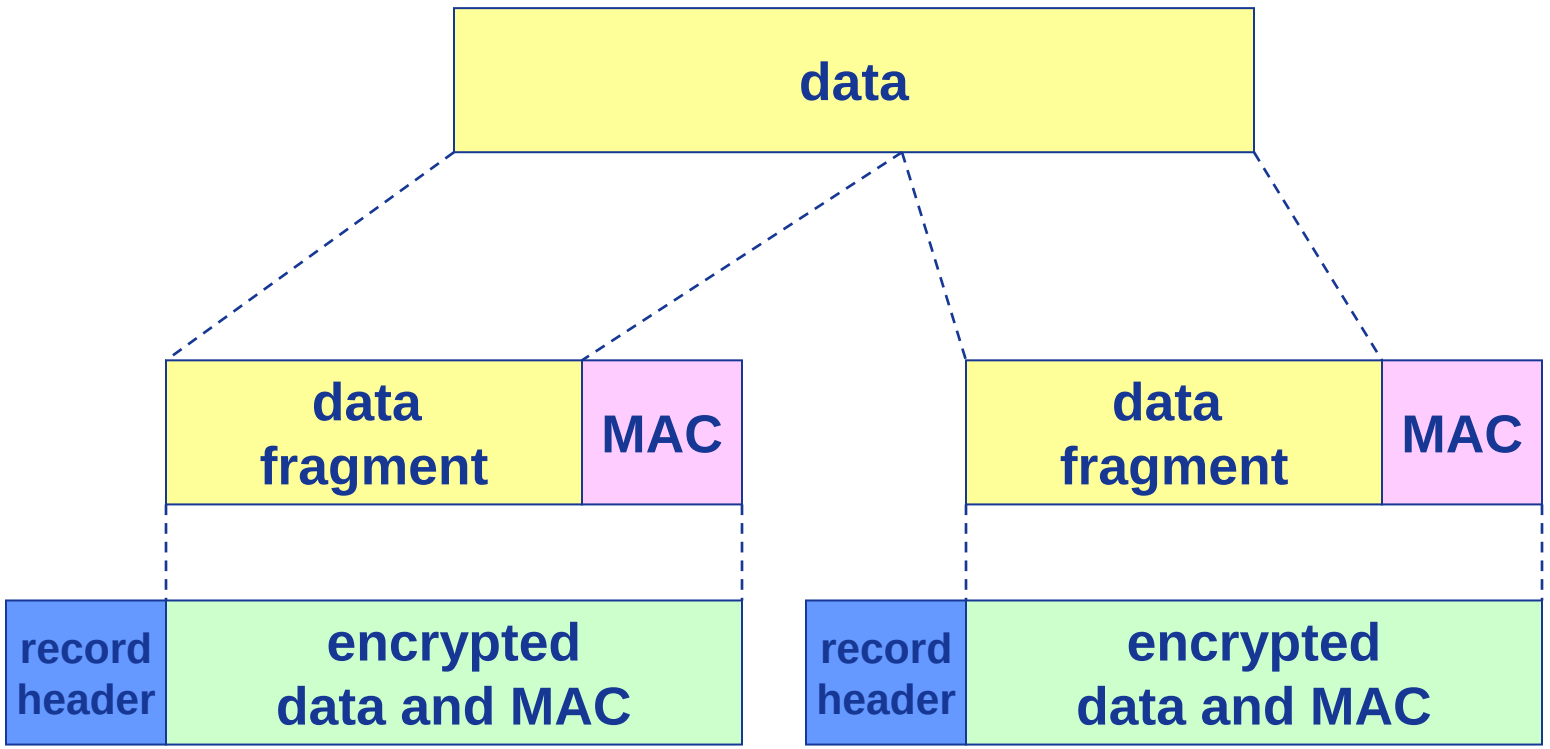
SSL握手过程(3)

- 为什么使用两个一次随机数？
- 假设Trudy嗅探Alice与Bob之间的所有报文
- 第二天，Trudy与Bob建立TCP连接，发送完全相同的记录序列
  - Bob(如Amazon)认为Alice对同一产品下发两个分离的订单
  - 解决方案: Bob为每次连接发送完全不同的一次随机数
    - 确保两天的加密密钥不同
  - Trudy的报文将无法通过Bob的完整性检验

SSL记录协议

- SSL记录协议的操作步骤：
  - 将数据分段成可操作的数据块
  - 对分块数据进行数据压缩
  - 计算MAC值
  - 对压缩数据及MAC值加密
  - 加入SSL记录头
  - 在TCP中传输

SSL记录协议



记录头(record header): 内容类型(ContentType); 版本; 长度

MAC: 包括序列号, MAC密钥 M_x

片段(fragment): 每个SSL片段为2^{14}字节 (~16KB)

SSL记录格式

数据和MAC是加密的(对称密钥加密算法)

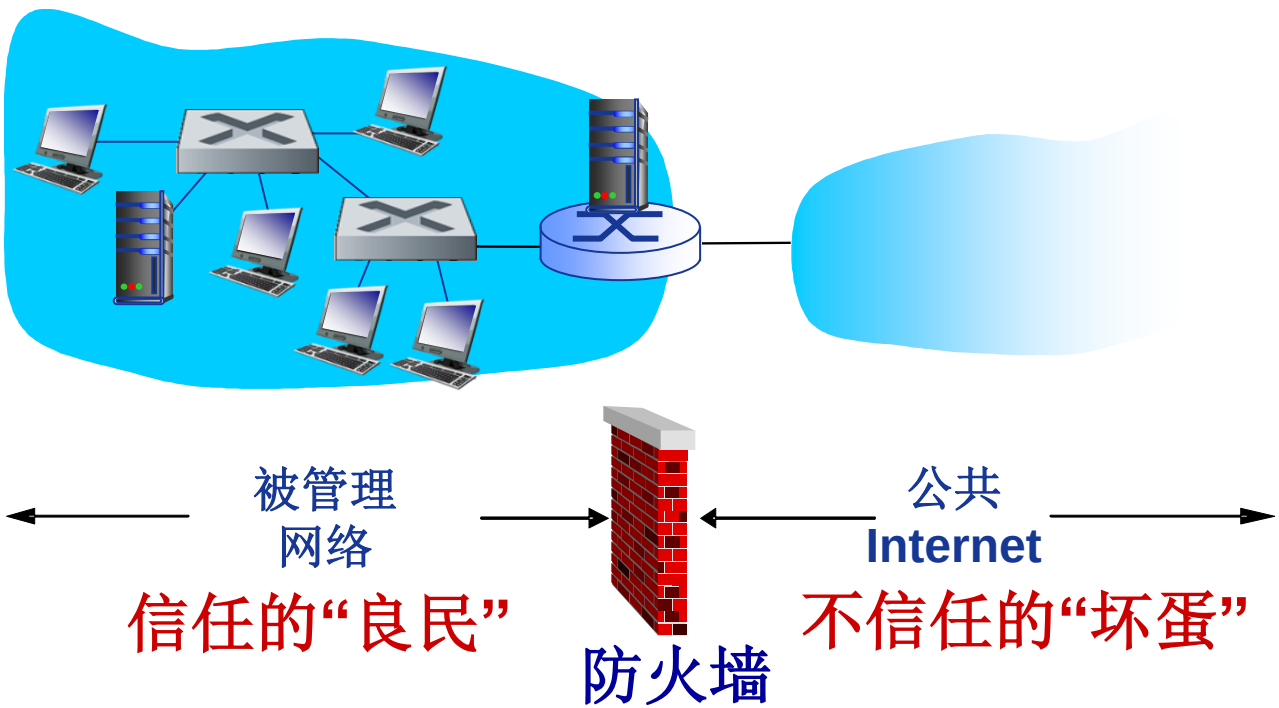
本讲主题

防火墙

防火墙

防火墙(firewall)

隔离组织内部网络与公共互联网，允许某些分组通过，而阻止其他分组进入/离开内部网络的软件/硬件设施。



为什么需要防火墙？

预防拒绝服务攻击（DoS）：

- SYN泛洪: 攻击者建立许多虚假TCP连接，耗尽资源，导致“真正”的连接无法建立

预防非法修改/内部数据访问:

- e.g., 攻击者替换CIA网站主页

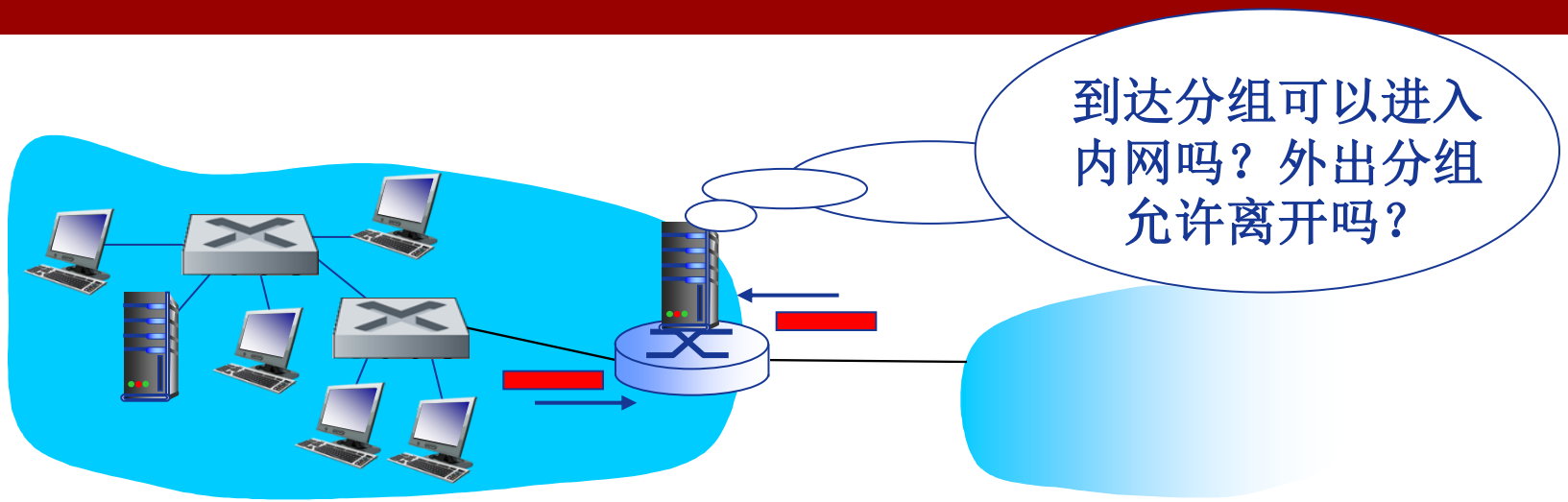
只允许对内部网络的授权访问：

- 认证的用户/主机

三种类型的防火墙:

- 无状态分组过滤器(stateless packet filters)
- 有状态分组过滤器(stateful packet filters)
- 应用网关(application gateways)

无状态分组过滤



- 内部网络通过路由器防火墙(router firewall)与Internet连接
- 路由器逐个分组过滤，决策是否转发/丢弃分组，依据：
  - 源IP地址、目的IP地址
  - TCP/UDP源、目的端口号
  - ICMP报文类型
  - TCP SYN和ACK标志位
  - ……

无状态分组过滤：举例

- 例1: 阻止协议字段=17，以及源或目的端口号=23的数据报进入与离开
  - 结果: 所有进入或离开的UDP流量，以及Telnet连接均被阻止
- 例2: 阻止进入的、 ACK=0的TCP段
  - 结果: 阻止外部客户与内部主机主动建立TCP连接，但是允许内部客户与外部主机主动建立连接

无状态分组过滤：更多例子

策略(Policy)	防火墙设置
不允许访问外部Web站点	丢弃所有目的端口号=80的外出分组
禁止进入的TCP连接，连接组织公共Web服务器除外	丢弃所有TCP SYN段，目的IP地址为130.207.244.203, 端口号为80的IP数据报除外
阻止Web电台应用，以防消耗可用带宽	丢弃所有进入的UDP分组，DNS分组和路由器广播分组除外
阻止你的网络被用于蓝精灵DoS攻击	丢弃所有发往广播地址(e.g. 130.207.255.255)的ICMP分组
阻止你的网络被路由跟踪	丢弃所有外出的TTL失效ICMP流量

访问控制列表

- **ACL(Access Control Lists):** 规则表，自顶向下应用于到达的分组：(action, condition)对

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----
deny	all	all	all	all	all	all

有状态分组过滤

- 无状态分组过滤器: 笨拙
  - 不加以区分放行满足条件的所有分组
  - 例如：放行dest port = 80、ACK=1的分组，即使没有建立的TCP连接:

action	source address	dest address	protocol	source port	dest port	flag bit
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK

- 有状态分组过滤器: 跟踪每个TCP连接
  - 跟踪连接建立(SYN)、拆除(FIN): 根据状态确定是否放行进入或外出的分组
  - 超时的非活动连接: 不再允许分组通过

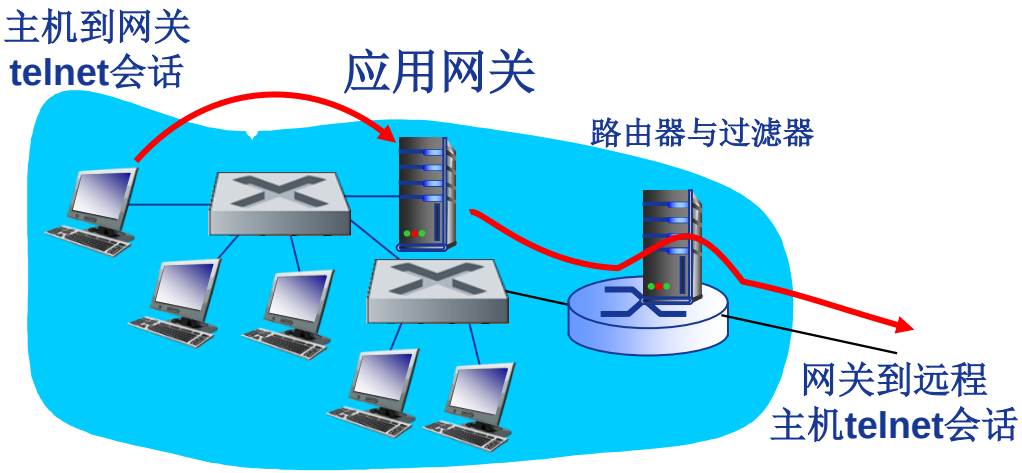
有状态分组过滤

- 扩展ACL，以便在放行分组前，检测连接状态表

action	source address	dest address	proto	source port	dest port	flag bit	check connection
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	---	
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	----	X
deny	all	all	all	all	all	all	

应用网关

- 基于应用数据以及IP/TCP/UDP头部字段过滤分组
- 例如：允许特定用户telnet外部网络



1. 要求所有Telnet用户通过网关Telnet外部网络；
2. 对于授权的用户，网关代理用户与目的主机建立Telnet连接，并且在两个连接之间进行数据中继；
3. 路由器阻止所有不是由网关发起的Telnet连接。