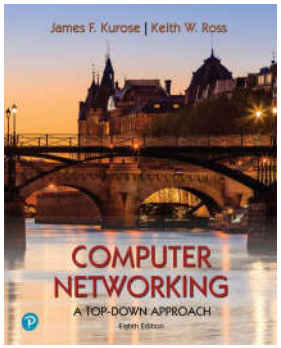
第 8 章
网络安全（上）

中国科学技术大学
自动化系 郑烇
改编自Jim kurose, Keith Ross



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

网络安全

本章目标：

- 网络安全原理:
  - 加密，不仅仅用于机密性
  - 认证
  - 报文完整性
- 网络安全实例：
  - 各个层次安全协议的例子
    - 应用层，传输层，网络层和链路层
  - 防火墙和入侵检测系统：IDS

第八章 提纲

- 什么是网络安全?
- 加密原理
- 认证，报文完整性
- 安全电子邮件
- 使TCP连接安全：TLS
- 网络层安全性：IPSec
- 无线和移动网络的安全
- 实践中的网络安全：防火墙和IDS

什么是网络安全?

机密性: 只有发送方和指定的接收方能否理解传输的报文内容
- 发送方加密报文
- 接收方解密报文

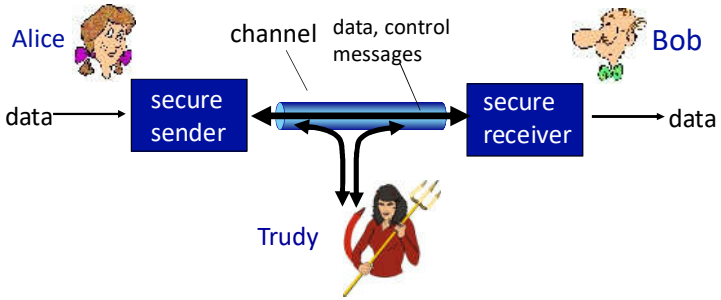
认证: 发送方和接收方需要确认对方的身份

报文完整性: 发送方、接受方需要确认报文在传输的过程中或者事后没有被改变

访问控制和服务的可用性: 服务可以接入以及对用户而言是可用的

朋友和敌人：Alice，Bob，Trudy

- 网络安全世界比较著名的模型
- Bob, Alice (lovers!) 需要安全的通信
- Trudy (intruder) 可以截获，删除和增加报文



2021中科大高网

朋友和敌人：Alice，Bob，Trudy

谁有可能是Bob, Alice?

- … 现实世界中的Bobs和Alices!
- 电子交易中的Web browser/server (e.g.,在线购买)
- 在线银行的client/server
- DNS 服务器
- BGP路由器做路由信息的交换
- 其它例子?

2021中科大高网

网络中存在这坏蛋!

Q: “bad guy”可以干什么?

A: 很多! (recall section 1.6)

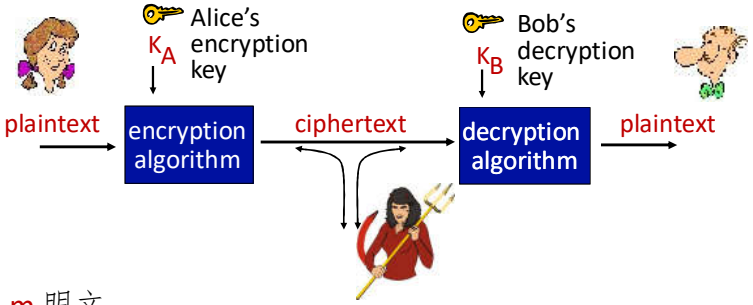
- *窃听：* 截获报文
- *插入：* 主动在连接上插入报文
- *伪装：* 可以在分组的源地址写上伪装的地址
- *劫持：* 将发送方或者接收方踢出，接管连接
- *拒绝服务：* 阻止服务被其他正常用户使用 (e.g.,通过对资源的过度使用)

2021中科大高网

第八章 提纲

- 什么是网络安全?
- 加密原理
- 认证，报文完整性
- 安全电子邮件
- 使TCP连接安全：TLS
- 网络层安全性：IPSec
- 无线和移动网络的安全
- 实践中的网络安全：防火墙和IDS

加密术语



m 明文

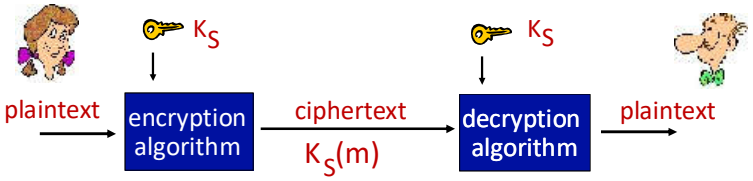
$K_A(m)$ 密文，采用K_A进行加密

$m = K_B(K_A(m))$

破解一个加密方案

- 唯密文攻击：Trudy仅有密文可供分析
- 2个方法：
  - 暴力破解：尝试所有可能的秘钥
  - 统计分析
- 已知明文攻击：Trudy 拥有部分与密文对应的明文
  - e.g.， 在单码替代密码中，Trudy已经知道某些字母之间的匹配a, l, i, c, e, b, o,
- 选择明文攻击：Trudy 可以得到选择明文对应的密文

对称秘钥加密



对称秘钥加密：Bob和Alice共享相同的秘钥（对称）：K

- 例如：单字母替换密码，key是替换模式

Q: Bob和Alice如何就秘钥的值达成一致呢?

简单对称加密方案

替换密码: 将一个字母换成另外一个字母

- 单码替换密码: 将一个字母替换成另外一个字母

```
plaintext:  abcdefghijklmnopqrstuvwxyz
ciphertext: mnbvcxzasdfghjklpoiuytrewq
```

e.g.: `Plaintext: bob. i love you. alice`
`ciphertext: nkn. s gktc wky. mgsbc`

加密秘钥: 26个字母和26字母的映射关系

一个更加复杂的加密方法

- n个替换密码，$M_1, M_2, \cdots, M_n$
- 循环模式：
  - e.g.， n=4：M_1, M_3, M_4, M_3, M_2； M_1, M_3, M_4, M_3, M_2；..
- 对于每个明文字符，采用循环替换模式的下一个
  - dog：d from M_1，o from M_3，g from M_4

加密的秘钥： n个替换模式（字母替换表）和循环模式

- 秘钥不仅仅是：字母之间的映射模式

对称密钥加密：DES

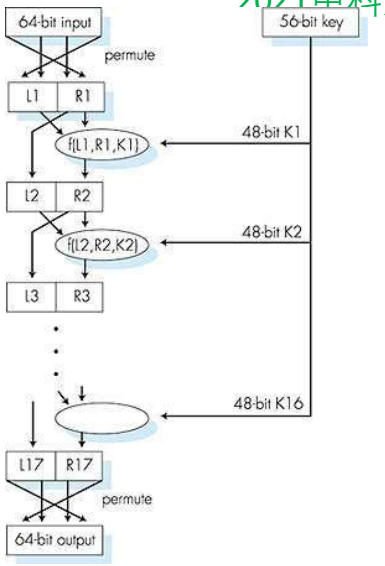
DES: Data Encryption Standard

- US 加密标准[NIST 1993]
- 56-bit 对称密钥，64-bit明文输入
- 块加密、密码块链
- DES有多安全?
  - DES挑战：56-bit密钥加密的短语（“Strong cryptography makes the world a safer place”）采用暴力解密，不到一天时间
  - 可能有后门?
- 使DES更安全：
  - 使用3个key、3重DES 运算
  - 密文分组成串技术

对称密钥加密：DES

DES: Data Encryption Standard

- 初始替换
- 16 轮一样的函数应用，每一轮使用的不同的48bit密钥
- 最终替换



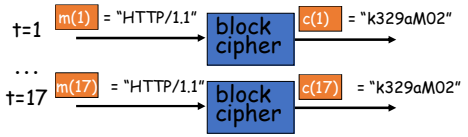
AES：Advanced Encryption Standard

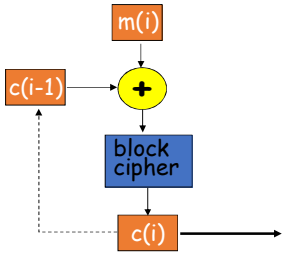
- 新的对称 密钥NIST标准(Nov. 2001) 用于替换 DES
- 数据128bit成组加密
- 128, 192, or 256 bit keys
- 暴力法解密如果使用1秒钟破解 DES，需要花149万亿年破解AES

密码块链

- 密码块：如果输入块重复，将会得到相同的密文块

- *密码块链：* 第i轮输入m(i)，与前一轮的密文，c(i-1)做异或





公开密钥密码

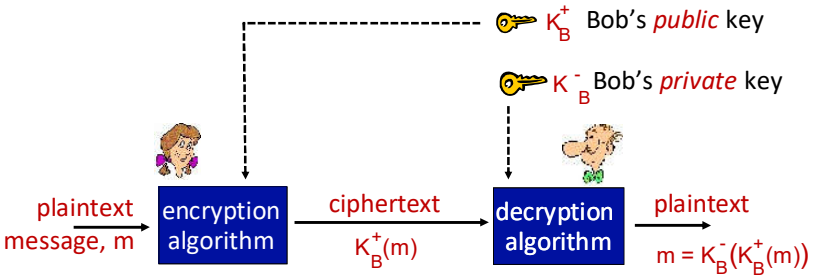
对称加密：

- 需要发送方和接收方对共享式对称密钥达成一致
- Q：但是他们如何第一次达成一致（特别是他们永远不可能见面的情况下）？

公开秘钥加密

- 完全不同的方法[Diffie-Hellman76, RSA78]
- 发送方和接收方无需共享密钥
- *一个实体的公钥公诸于众*
- *私钥*只有接收方自己知道

公开密钥加密



Wow -公钥密码学彻底改变了2000年（以前只有对称密钥）的密码学！
- 同样的想法几乎同时，且独立地出现在美国和英国

公开密钥加密算法

要求：

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that
$$K_B^-(K_B^+(m)) = m$$

2. 给定公钥K_B^+，导出私钥K_B^-计算上不可行

RSA: Rivest, Shamir, Adelson algorithm

预先知识：模运算

- x mod n = x除以n得到的余数
- 一些事实：
  [(a mod n) + (b mod n)] mod n = (a+b) mod n
  [(a mod n) - (b mod n)] mod n = (a-b) mod n
  [(a mod n) * (b mod n)] mod n = (a*b) mod n
- 因此
  $(a \bmod n)^d \bmod n = a^d \bmod n$
- 例子：x=14, n=10, d=2:
  $(x \bmod n)^d \bmod n = 4^2 \bmod 10 = 6$
  $x^d = 14^2 = 196 \quad x^d \bmod 10 = 6$

RSA：准备

- 报文：就是一个bit模式
- Bit模式可以唯一的用整数来表示
- 因此，加密一个报文就等于加密一个整数

例子：

- m= 10010001. 报文可以唯一的用十进制数145来代替.
- 为了加密m，我们加密相应的整数，得到一个新的整数(密文)

RSA：创建公钥/私钥对

1.选择2个很大的质数 p, q . （例如：每个1024 位）

2.计算 $n = pq$, $z = (p-1)(q-1)$

3.选择一个 e (要求 $e<n$) 和z 没有一个公共因子，互素 ("relatively prime").

4.选择 d 使得 $ed-1$ 正好能够被z整除.
(也就是: $ed \bmod z = 1$).

5.公钥 $\underbrace{(n,e)}_{K_B^+}$. 私钥 $\underbrace{(n,d)}_{K_B^-}$.

RSA：加密和解密

0. 给定按照上述算法得到的 (n,e) 和 (n,d)

1. 加密报文m $(<n)$, 加密算法
   $c = m^e \bmod n$

2.对接收到的密文c解密，如此计算
   $m = c^d \bmod n$

magic happens! $m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$

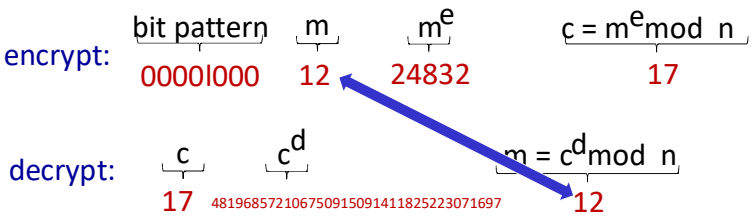
RSA例子:

Bob 选择 $p=5$, $q=7$. 因此 $n=35$, $z=24$.

$e=5$ (so e, z 互素).

$d=29$ (so $ed-1$ 能够被 z整除).

加密 8-bit报文



为什么RSA可行? $m = (m^e \bmod n)^d \bmod n$

一个有用的数论定理: 如果 p,q 都是素数，$n = pq$, *那么*:

$$x^y \bmod n = x^{y \bmod (p-1)(q-1)} \bmod n$$

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{ed \bmod (p-1)(q-1)} \bmod n$$

(使用上述定理)

$$= m^1 \bmod n$$

(因为我们选择ed 使得正好被z 除余1)

$$= m$$

RSA: 另外一个重要的特性

下面的特性将在后面非常有用:

$$\underbrace{K_B^-(K_B^+(m))}_{\text{先用公钥，然后用私钥}} = m = \underbrace{K_B^+(K_B^-(m))}_{\text{先用私钥，然后用公钥}}$$

结果一致!

为什么 $K_B^-(K_B^+(m)) = m = K_B^+(K_B^-(m))$?

直接遵循求模的运算法则:

$$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$$

$$= m^{de} \bmod n$$

$$= (m^d \bmod n)^e \bmod n$$

为什么RSA安全?

- 假设知道了Bob的公钥(n,e)，求解私钥d的难度有多大(n,d)?
- 本质上是在不知道2个因子p和q的情况下，求解n的因子
  - 事实是：分解一个大的整数是非常困难的

实践中的RSA：会话秘钥

- 在RSA中求幂次方的运算代价非常大
- DES（对称加密）最起码比RSA快100倍
- 采用公开秘钥加密体系创建一个安全的连接，然后建立第二个对称秘钥key，用于实际加密数据

会话秘钥, K_S

- Bob 和Alice采用RSA交换对称秘钥 K_S
- 一旦拥有了K_S, 采用对称加密来进行加密和解密

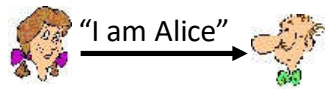
第八章 提纲

- 什么是网络安全?
- 加密原理
- 认证，报文完整性
- 安全电子邮件
- 使TCP连接安全：TLS
- 网络层安全性：IPSec
- 无线和移动网络的安全
- 实践中的网络安全：防火墙和IDS

认证

目标: Bob需要Alice“证明”她的身份

Protocol ap1.0: Alice 说“I am Alice”

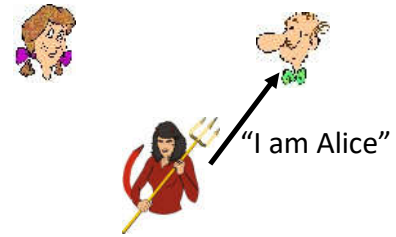


failure scenario??

认证

目标: Bob需要Alice“证明”她的身份

Protocol ap1.0: Alice 说“I am Alice”



在网络中，Bob看不到Alice，所以Trudy简单申明其是 Alice

认证：新的尝试

目标: Bob需要Alice“证明”她的身份

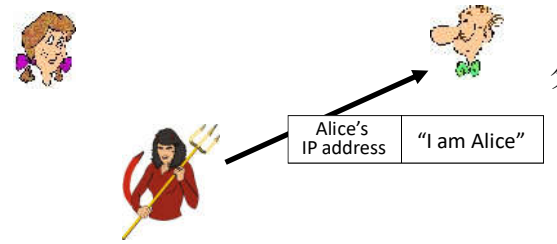
Protocol ap2.0: Alice 说“I am Alice”，具备Alice的IP地址

failure scenario??

认证：新的尝试

目标: Bob需要Alice“证明”她的身份

Protocol ap2.0: Alice 说“I am Alice”，具备Alice的IP地址

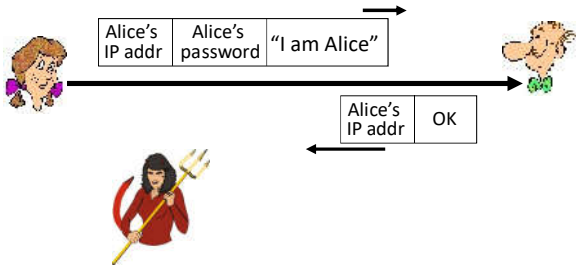


Trudy 可以创建一个分组伪造Alice的IP地址

认证:第三次尝试

目标: Bob需要Alice“证明”她的身份

Protocol ap3.0:Alice说“I am Alice”，而且传送她的密码来证明

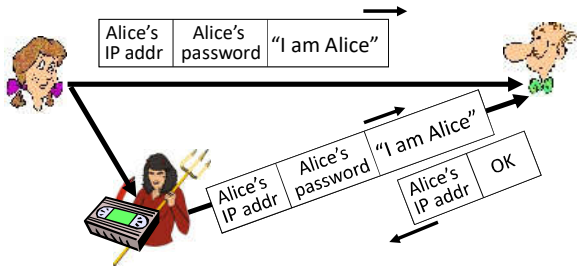


failure scenario??

认证:第三次尝试

目标: Bob需要Alice“证明”她的身份

Protocol ap3.0: Alice说“I am Alice”，且传送她的密码来证明

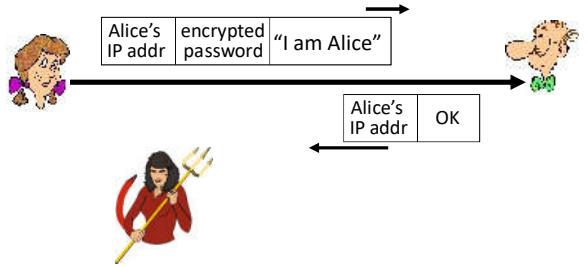


重放攻击 playback attack: Trudy记录 Alice的分组，事后向Bob重放

认证:升级的第三次尝试

目标: Bob需要Alice“证明”她的身份

Protocol ap3.0: Alice 说 “I am Alice” ，而且传送她的加密之后的密码来证明

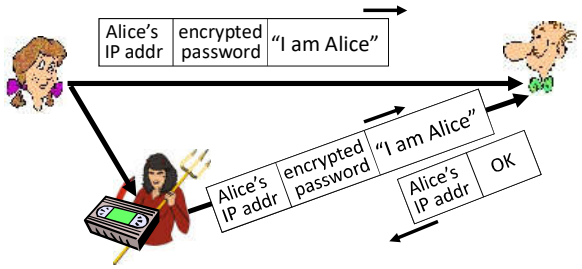


failure scenario??

认证:升级的第三次尝试

目标: Bob需要Alice“证明”她的身份

Protocol ap3.0: Alice 说 “I am Alice” ，而且传送她的加密之后的密码来证明



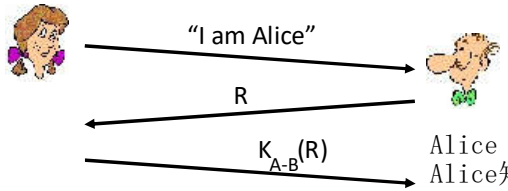
重放攻击仍然有效: Trudy 记录records Alice的分组，然后过后重放给Bob

认证: 第四次尝试

目标：避免重放攻击

Nonce: 一个协议生存期只用一次的整数 (R)

protocol ap4.0: 为了证明Alice的活跃性，Bob发送给Alice一个nonce R. Alice返回加密之后的R，使用双方约定好的key



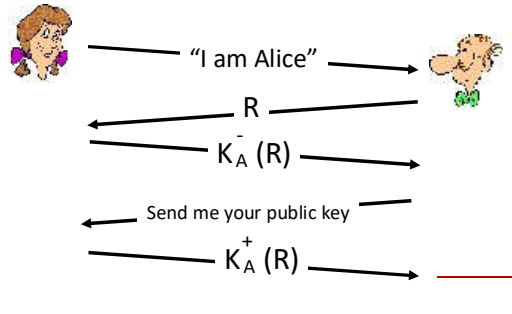
Alice 是活跃的，只有Alice知道这个加密的nonce，因此一定是Alice!

Failures, drawbacks?

认证: ap5.0

ap4.0 需要双方共享一个对称式的密钥-是否可以通过公开密钥技术进行认证呢?

ap5.0:使用nonce R,公开密钥加密技术



Bob 计算

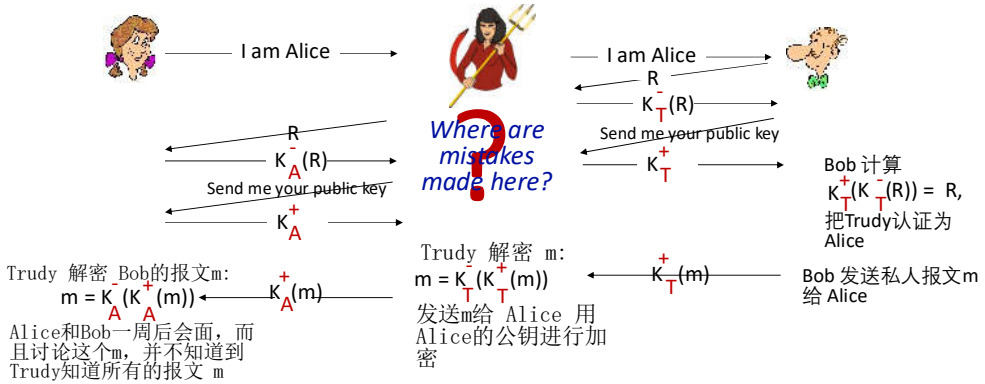
$K_A^+(K_A^-(R)) = R$

可以知道只有 Alice 拥有该私钥，才能得到$K_A^-(R)$

$K_A^+(K_A^-(R)) = R$

认证: ap5.0 – 安全漏洞!

中间攻击: Trudy 在 Alice (to Bob)和 Bob之间 (to Alice)



认证: ap5.0 – 安全漏洞!

中间攻击: Trudy 在 Alice (假装成 Bob)和 Bob之间 (假装成Alice)

难以检测:

- Bob收到了Alice发送的所有报文，反之亦然. (e.g., so Bob, Alice一个星期以后相见，回忆起以前的会话)
- 问题是Trudy也接收到了所有的报文!

- 本质：Bob没有拿到真正Alice的公钥
- 如果可保证拿到Alice的公钥，那么采用公开密钥体系就行认证也是可行的

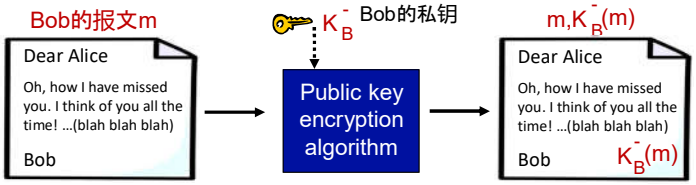
第八章 提纲

- 什么是网络安全?
- 加密原理
- 认证，报文完整性
- 安全电子邮件
- 使TCP连接安全：TLS
- 网络层安全性：IPSec
- 无线和移动网络的安全
- 实践中的网络安全：防火墙和IDS

数字签名

数字签名类似于手写签名：

- 发送方 (Bob) 数字签署了文件，前提是他(她)是文件的拥有者/创建者.
- 可验证性，不可伪造性: 接收方 (Alice)可以向其他人证明是一定Bob而不是其他人（包括Alice自己）签署了该报文
- 报文m的简单数字签名：
  - Bob使用其私钥对m进行了签署，创建数字签名的 $K_B^-(m)$



数字签名

- 假设Alice收到报文m, 以及数字签名$K_B^-(m)$

Alice 使用Bob的公钥K_B^+对$K_B^-(m)$进行验证， 判断 $K_B^+(K_B^-(m)) = m$是否成立？

- 如 $K_B^+(K_B^-(m)) = m$成立, 那么签署这个文件的人一定拥有Bob的私钥.

Alice 可以验证:

- ✓ Bob 签署了m.
- ✓ 不是其他人签署了m.
- ✓ Bob签署了m 而不是m'.

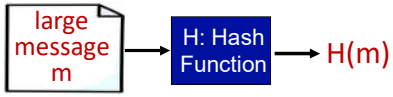
不可抵赖性:

- ✓ Alice可以拿着m,以及数字签名$K_B(m)$到法庭上，来证明是Bob签署了这个文件 m.

报文摘要

对长报文进行公开密钥加密算法的运行需要耗费大量的时间

Goal: 固定长度，容易计算的 “指纹（fingerprint）*对m使*用散列函数H，获得固定长度的报文摘要*H(m)*



散列函数的特性:

- 多对1
- 结果固定长度
- 给定一个报文摘要x，反向计算出原报文在计算上是不可行的x = H(m)

Internet校验和：弱的散列函数

Internet 校验和拥有一些散列函数的特性:

- 产生报文m的固定长度的摘要 (16-bit sum)
- ✓多对1的

但给定一个散列值，很容易计算出另外一个报文具有同样散列值：

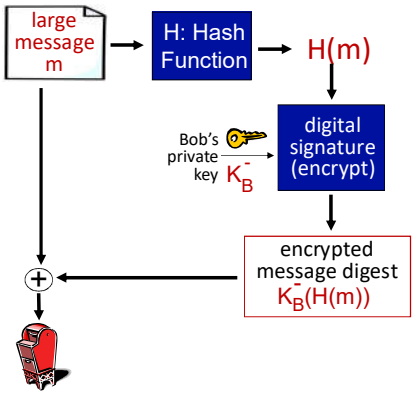
message	ASCII format
I O U 1	49 4F 55 31
0 0 . 9	30 30 2E 39
9 B O B	39 42 D2 42
	B2 C1 D2 AC

message	ASCII format
I O U 9	49 4F 55 39
0 0 . 1	30 30 2E 31
9 B O B	39 42 D2 42
	B2 C1 D2 AC

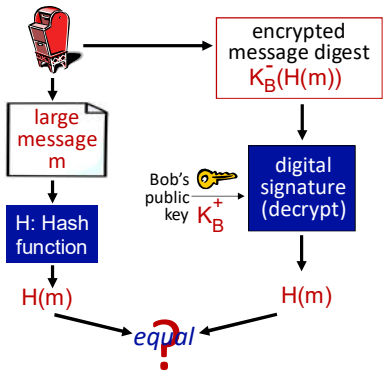
不同的报文
但是相同的校验和!

数字签名 = 对报文摘要进行数字签署

Bob 发送数字签名的报文：



Alice校验签名和报文完整性：

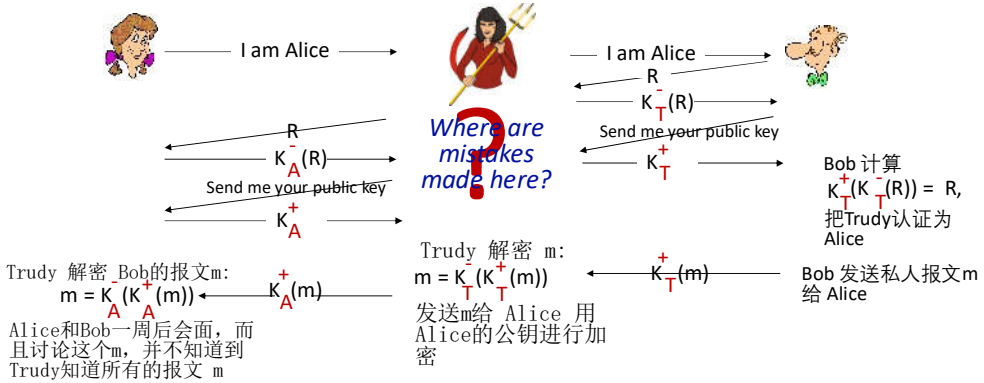


散列函数算法

- MD5散列函数(RFC 1321)被广泛地应用
  - 4个步骤计算出128-bit的报文摘要
  - 给定一个任意的128-bit串x, 很难构造出一个报文m具有相同的MD5摘要
- SHA-1也被使用
  - US标准 [NIST, FIPS PUB 180-1]
  - 160-bit报文摘要

修复ap5.0的安全漏洞

回顾中间攻击的问题：Trudy 站在Alice (假扮Bob)和 Bob (假扮Alice)之间

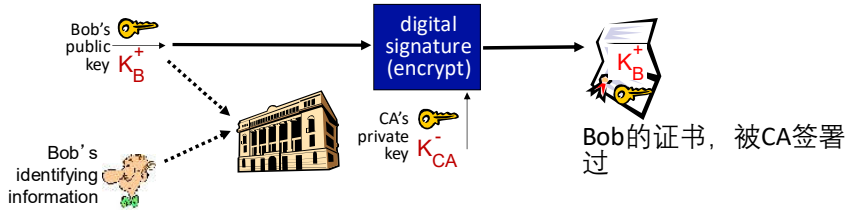


需要认证的公钥

- 动机：Trudy对Bob做一个披萨恶作剧
  - Trudy创建一个e-mail订单：亲爱的披萨店，*请送4个辣味意大利香肠披萨给我，谢谢，Bob*
  - Trudy采用自己的私钥签署了这个订单
  - Trudy发送这个订单给披萨店
  - Trudy将自己的公钥发给披萨店，说这是Bob的公钥
  - Pizza店验证该签名；然后发了4个辣味意大利香肠披萨给Bob
  - Bob甚至不喜欢辣味意大利香肠披萨

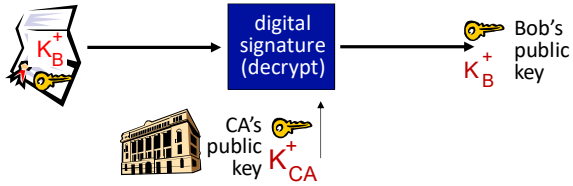
公钥认证中心 (CA)

- certification authority (CA): 将一个公钥和拥有它的实体E做个捆绑
- E (person, router) 到CA那里注册他的公钥， E 提供给CA，自己身份的证据 “proof of identity”
  - CA创建一个证书，捆绑了E实体信息和他的公钥.
  - Certificate包括了E的公钥，而且是被CA签署的（被CA用自己的私钥加了密的） – CA说 “this is E’s public key



公钥认证中心 (CA)

- 当Alice需要拿到Bob公钥:
  - 获得Bob的证书certificate (从Bob或者其他地方)
  - 使用CA的公钥来验证 Bob的证书
    - 前提需要可靠地获得CA的公钥，CA的证书
    - CA证书获得带外方式：安装系统带的，或者直接信赖的这个公钥
    - CA证书一般是自己用自己的私钥签署自己的公钥



互联网的信任树

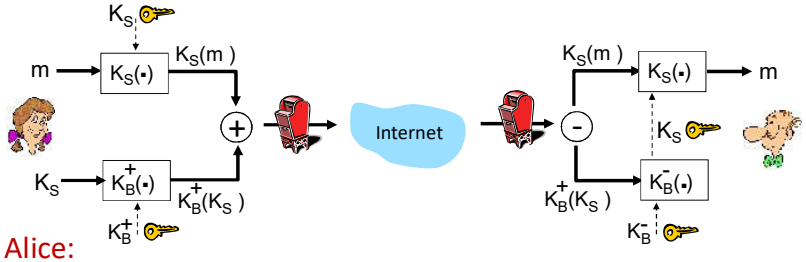
- 根证书：根证书是未被签名的公钥证书或自签名的证书
  - 拿到一些CA的公钥
  - 渠道：安装OS自带的数字证书；从网上下载，你信任的数字证书
- 信任树：
  - 信任根证书CA颁发的证书，拿到了根CA的公钥
    - 信任了根
  - 由根CA签署的给一些机构的数字证书，包含了这些机构的数字证书
  - 由于你信任了根，从而能够可靠地拿到根CA签发的证书，可靠地拿到这些机构的公钥

第八章 提纲

- 什么是网络安全?
- 加密原理
- 认证，报文完整性
- 安全电子邮件
- 使TCP连接安全：TLS
- 网络层安全性：IPSec
- 无线和移动网络的安全
- 实践中的网络安全：防火墙和IDS

安全电子邮件：机密性

Alice需要发送机密的报文m给Bob.

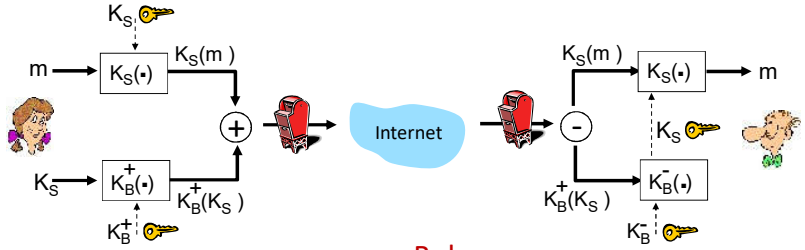


Alice:

- 产生随机的对称密钥，K_S.
- 使用K_S对报文加密(为了效率)
- 对 K_S 使用 Bob的公钥进行加密.
- 发送$K_S(m)$ 和$K_B(K_S)$ 给 Bob.

安全电子邮件：机密性（续）

Alice需要发送机密的报文m给Bob.

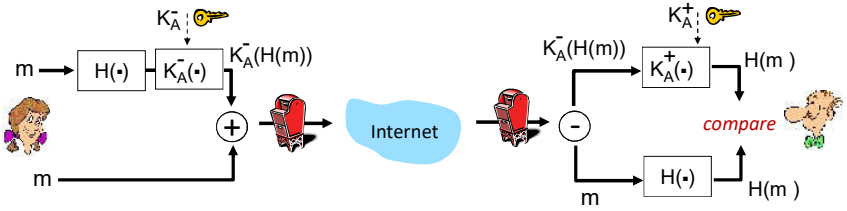


Bob:

- 使用自己的私钥解密 K_S
- 使用 K_S 解密 $K_S(m)$ 得到报文

安全电子邮件：完整性和可认证性

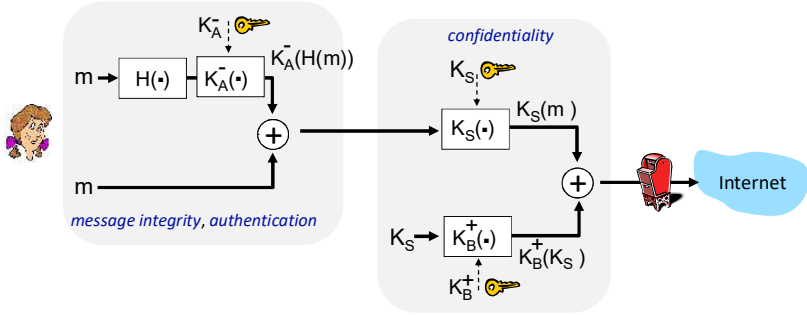
•Alice 需要提供源端的报文完整性和可认证性



•Alice 数字签署文件(采用自己的私钥签署报文的散列)
•发送报文（明文）和 数字签名

安全电子邮件：私密性、完整性和可认证性

Alice 需要提供机密性，源端可认证性和报文的完整性



Alice用了3个keys：自己的私钥，Bob的公钥，新产生的对称式密钥

Bob端的对应动作是?