



# 网络安全总复习

董建阔

南京邮电大学



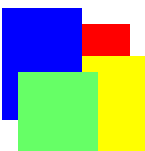
# 主要内容

- 网络安全概述
- 数据加密导论\*
- 真实性验证技术\*
- 访问控制技术\*
- 网络攻击防御
- 网络安全加固\*
- 网络安全应用



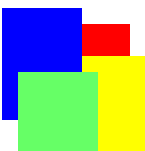
# 网络安全概述

- 网络安全定义\*
- 网络安全目标
- 网络安全内涵\*
- 网络安全风险
- 网络安全技术组成\*
- 网络安全关键技术



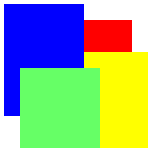
# 网络安全定义

- 网络安全是在通信安全、计算机安全和密码技术的基础上建立的一种网络环境下的安全可控技术体系，其目的是保护网络系统以及网络应用系统(包含网络及其应用系统内传递和存储数据)的保密性、完整性和可用性。
- 网络安全不同于通信安全和计算机安全技术，网络安全涉及在网络环境下的应用系统安全性，以及网络系统自身的安全性。



# 网络安全目标

- 网络安全目标涉及两个方面: 网络系统本身的安全性, 网络应用系统的安全性.
- 安全性可以进一步分解成保密性、完整性和可用性。
- 保密性: 网络及应用系统内传递数据的保密性, 例如防范截获并窃取数据; 网络及应用系统结构以及配置的保密性, 例如防止通过嗅探报文得到网络结构信息.
- 完整性: 网络及应用系统内传递数据的完整性, 例如防范数据传递途中被篡改; 网络及应用系统结构以及配置的完整性, 例如不会被篡改配置.
- 可用性: 网络及应用系统功能以及对外提供服务的可用性; 包括网络及应用系统存储的数据可用性



# 网络安全技术组成

- 网络安全技术包括真实性验证、访问控制、攻击防御、网络安全加固以及网络安全应用。
- 虽然数据加密在网络安全技术中也扮演一个十分关键的、不可缺少的角色，数据加密并不是网络安全研发的技术。

第3层	网络安全加固		网络安全应用	
第2层	真实性验证	访问控制		攻击防御
第1层	数据加密			

网络安全技术的组成



# 网络安全的几种风险形式

- 网络安全技术是应对不同网络安全风险而发展起来的技术。
- 目前网络系统存在多种安全风险，例如假冒用户或假冒IP地址、窃听和篡改网络传递的数据、重播网络报文、中断网络服务、网络蠕虫攻击等。
- 研究和分析这些网络安全风险模型, 才能设计出较为完整的网络安全控制体系



# 应对网络威胁的网络安全技术

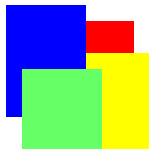
- 网络安全技术是针对网络安全威胁设计的一套安全保护机制。
- 网络真实性验证技术，通过对网络实体身份和数据的真实性验证，可以防范假冒和篡改型网络威胁，
- 网络访问控制技术，通过对访问权限的控制，可以防范窃听型网络威胁和渗透类网络攻击，
- 网络攻击防御技术可以检测和防御渗透类网络攻击以及拒绝服务攻击。



# 数据加密导论

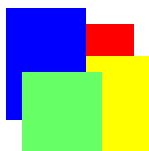
---

- 数据加密基本概念
- 传统数据加密概述\*
- 公钥数据加密概述\*



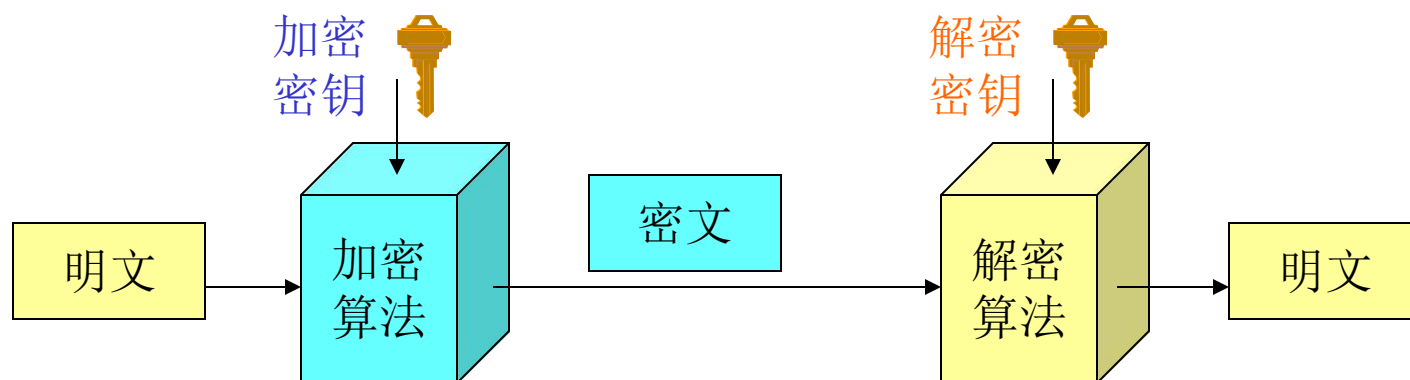
# 数据加密基本概念

- 数据加密基本概念
- 密码破译技术
- 加密系统的安全性
- 现代数据加密分类



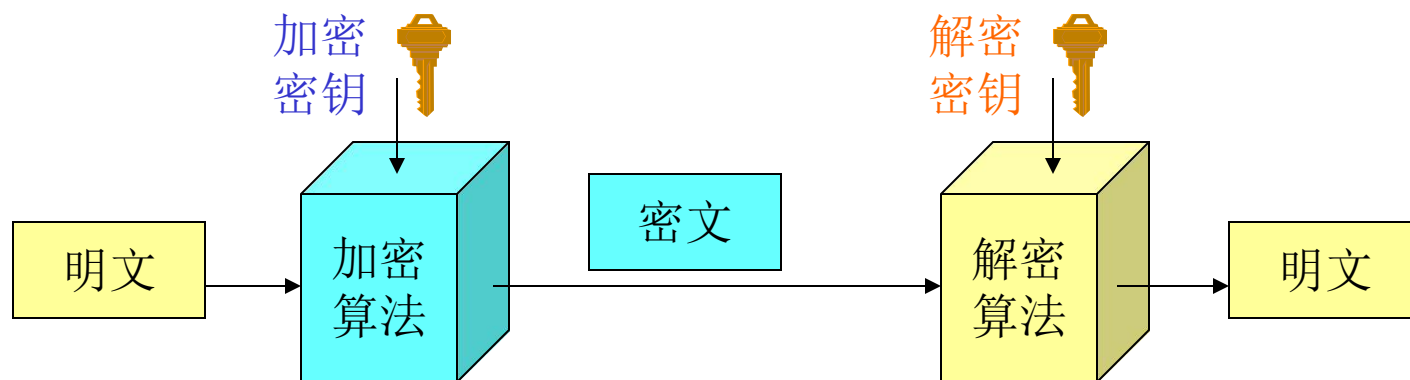
# 数据加密基本概念

- 明文(Plaintext, P): 对信息加密前的数据。
- 密文(Ciphertext, C): 对信息加密后的数据。
- 加密算法(E): 将明文转换为密文的处理过程。



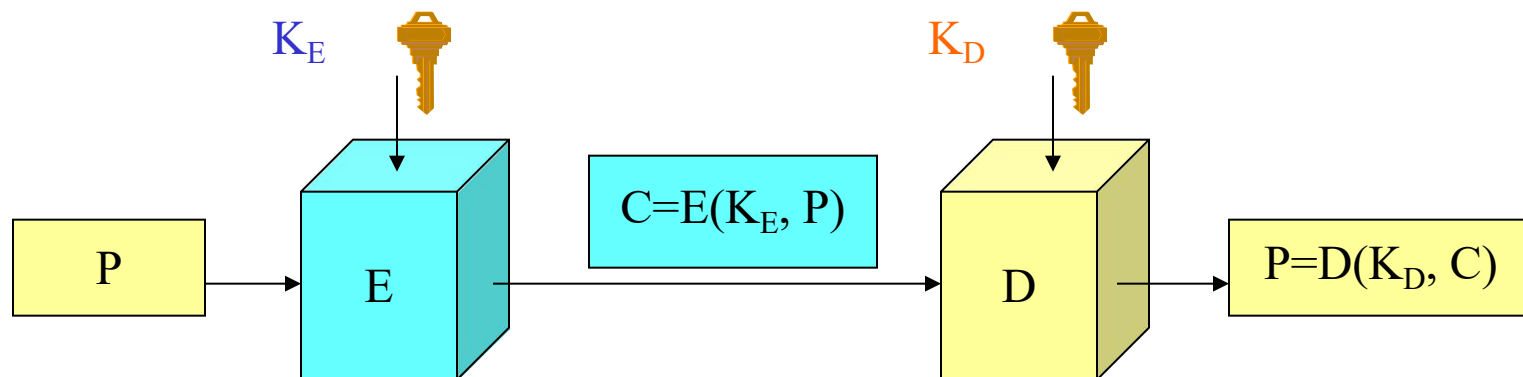
# 数据加密基本概念(续)

- 解密算法(D): 将密文转换为明文的处理过程
- 加密密钥( $K_E$ ): 控制加密处理的、包含特定信息的数据
- 解密密钥( $K_D$ ): 控制解密处理的、包含特定信息的数据



# 数据加密中符号表示

- 加密算法:  $C = E(K_E, P) = K_E \{P\}$
- 解密算法:  $P = D(K_D, C) = D(K_D, E(K_E, P))$
- 从以上加密和解密的公式可以看出, 经过数学抽象表示的加密和解密过程更加简洁、明了, 便于梳理思路, 掌握本质内容。



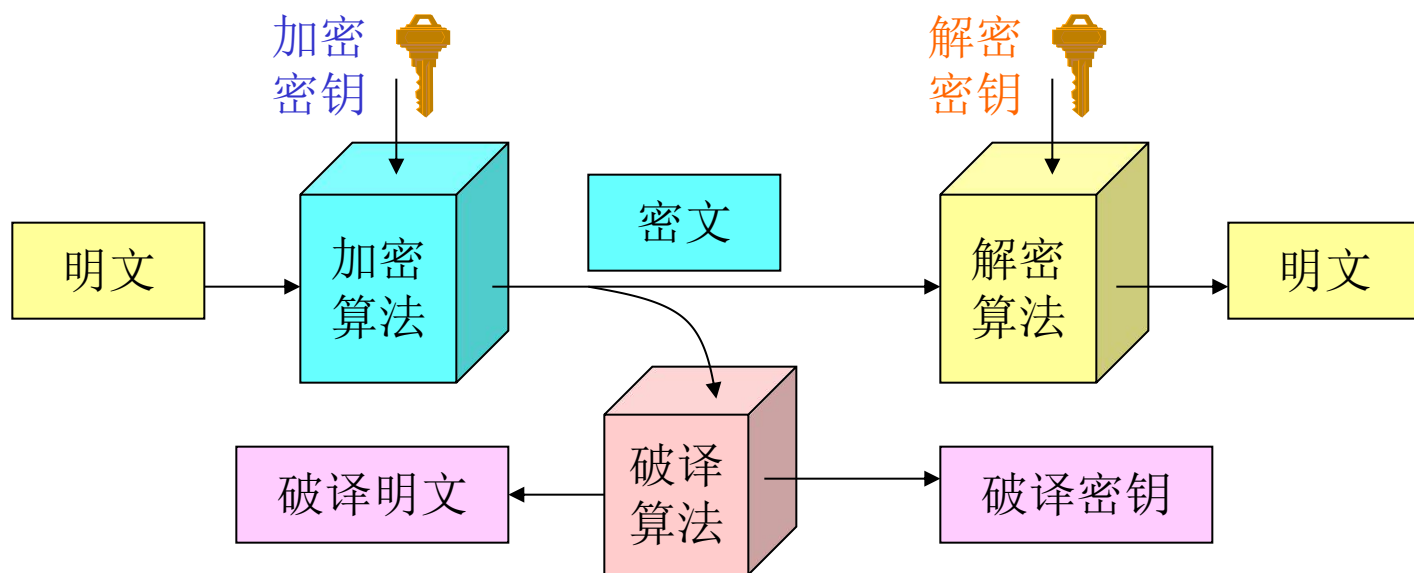


# 密码破译技术

- 安全是指在某种环境和条件下相对某种风险模型的安全。所以，必须根据目前常用的密码破译技术设计和评测加密算法。
- Diffie和Hellman在1976年罗列了3种密码分析(也称为密码攻击)方式
  - 已知密文攻击
  - 已知明文攻击
  - 选择明文攻击

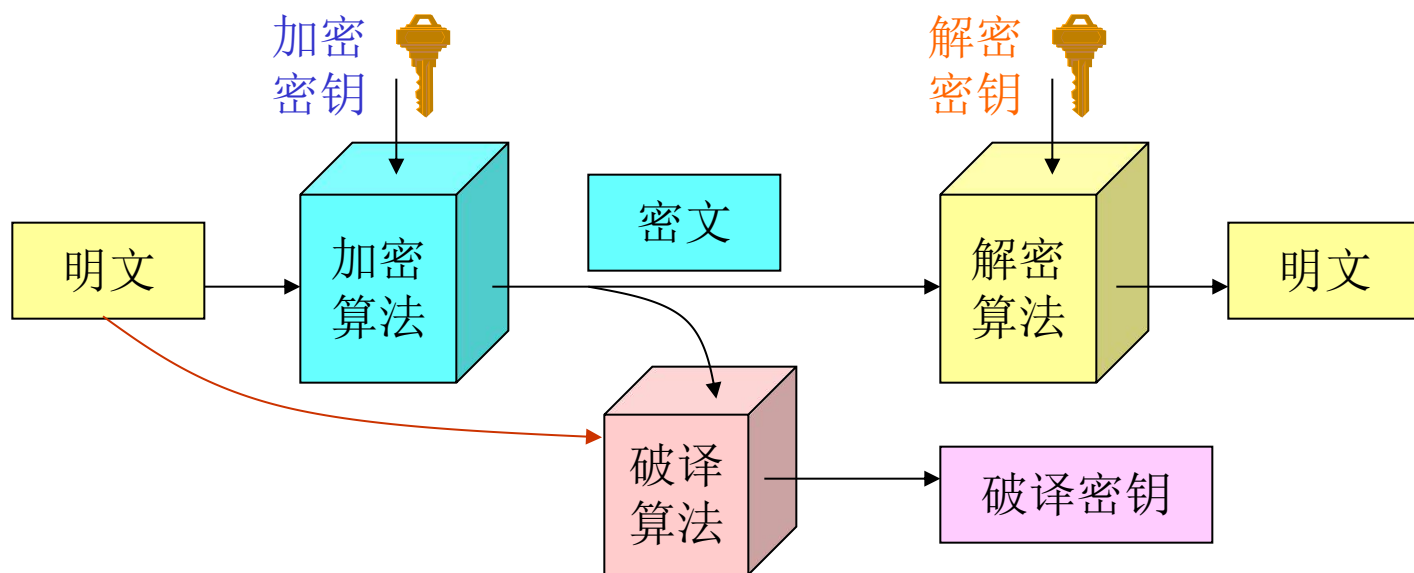
# 已知密文攻击

- 已知密文:攻击者仅仅掌握密文,试图破译对应的明文、加密算法和密钥。



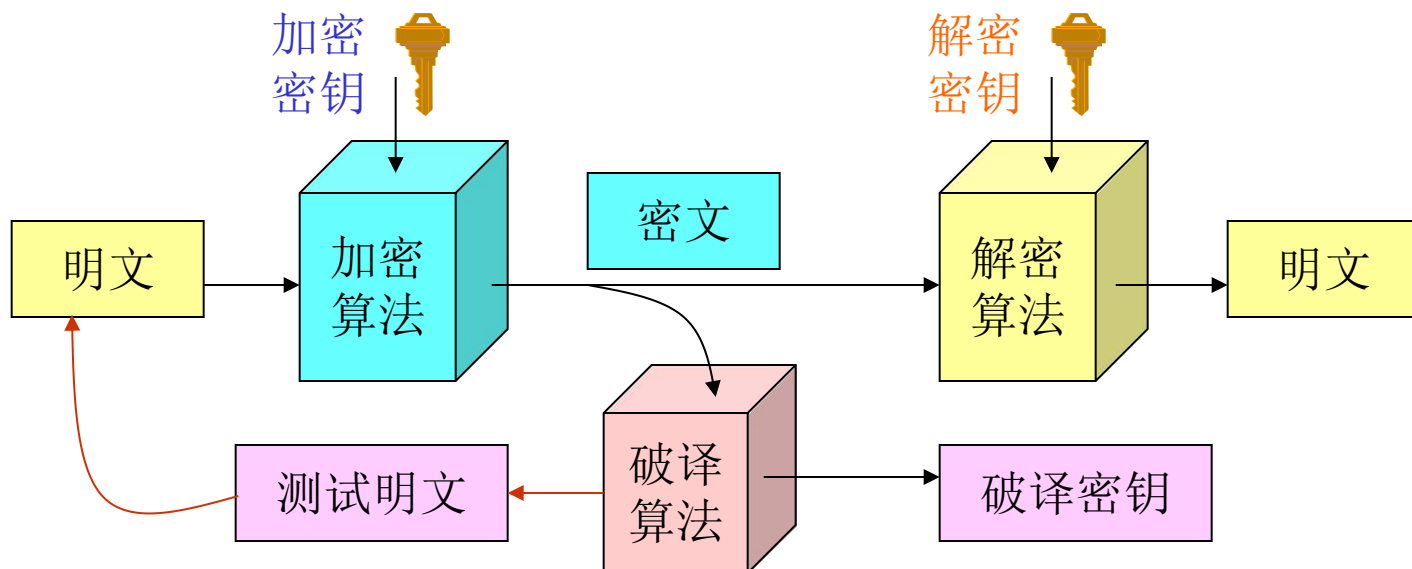
# 已知明文攻击

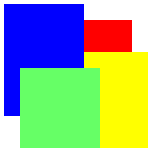
- 已知明文:攻击者掌握了大量明文和对应的采用相同加密算法和密钥产生的密文, 试图破译加密这些密文的算法和密钥。这是一种获取对应明文的攻击。



# 选择明文攻击

- 选择明文:攻击者可以向被攻击的加密系统提交多个选择的明文并可以获取对应生成的密文, 试图破译加密系统采用的加密算法和密钥。属于有针对性攻击。





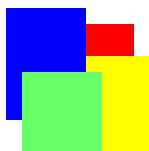
# 安全加密系统的定义

- 已知明文攻击属于“被动系统识别”类攻击，而选择明文攻击属于“主动系统识别”类攻击。
- 作为一个安全的加密系统应该是一类难以识别的系统，至少必须防范“被动系统识别”（已知明文）类攻击，最好能够防范“主动系统识别”（选择明文）类攻击。



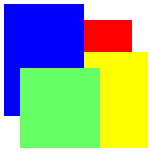
# 加密系统的安全性

- W. Diffie和M. Hellman将加密系统的安全性分成两种类型：
- 相对安全(或计算安全)的加密系统
  - 该类系统由于破译者的计算成本限制或者计算能力限制而看作是安全的。如果破译者**不考虑计算成本**，或者由于**计算技术发展**使得计算能力有大幅度提高，则这类系统就**需要重新**进行安全评估。
- 绝对安全(或无条件安全)的加密系统
  - 无论攻击者花费多少时间、使用多么高级的计算技术都无法破译的加密系统。



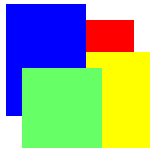
## 加密系统的安全性(续1)\*

- 一种可以证明的无条件安全加密系统是“一次性覆盖数”系统。由于计算成本过高而无法实用。
  - 该系统设计的密钥必须与明文同样长度，通过该密钥与明文进行“异或”操作，完成对明文的加密。该加密系统必须保证“一次一密钥”，即对于不同的明文采用不同的加密密钥。
- 目前实际可行的、并且得到广泛应用的还是相对安全(计算安全)的加密系统。
- 为了对于计算安全加密系统有一个量化的概念，需要了解一些典型常数和参数的数量级别。



# 现代数据加密分类

- 数据加密有多种分类方法，例如可以根据加密数据过程中对明文的处理方式，分成块加密方法和流加密方法；
- 也可以根据加密和解密数据过程中采用的密钥是否相同，分成对称密钥加密方法(相同)和不对称密钥加密方法(不同)。
- 现代数据加密本质上可以分成两个部分：传统数据加密，公钥数据加密

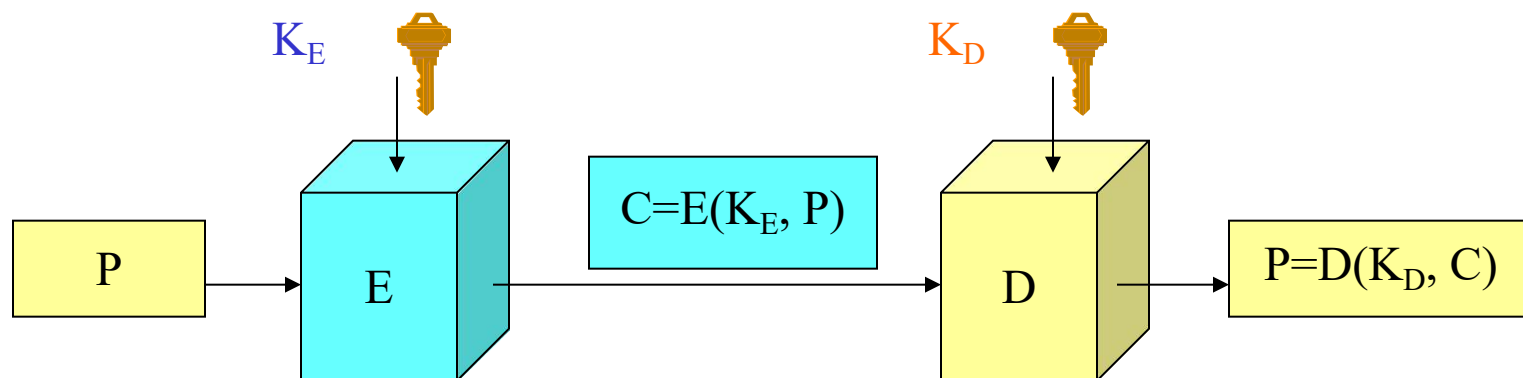


# 流加密方法与块加密方法

- 流加密方法是指连续对数据流中的某个较小的数据单元进行简单运算生成密文流的方法。
  - 这里的较小数据单元可能是1个八位位组（即一个8比特长度的字节）或者2个八位位组。
  - 数据流加密要求快速，满足话音流和视频流的短时延需求。
- 块加密方法是指对某个固定长度的数据块进行一系列复杂的运算生成相同长度密文块的方法。
  - 通常采用的固定长度是64个比特，现在建议采用128个比特。

# 对称密钥与不对称密钥加密方法

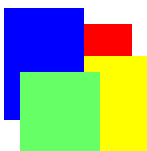
- 对称密钥加密方法是指加密和解密过程都采用相同的密钥，即在下图中 $K_E = K_D$ 。
- 不对称密钥加密方法是指加密和解密过程采用不同的密钥，即在下图中 $K_E \neq K_D$ 。





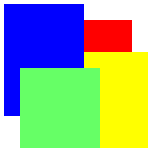
# 传统数据加密

- 对称密钥, 单个密钥进行加密和解密,
- 密钥保密才能保证密文保密
- 优点: 加密和解密运算简单、高效
- 缺点: 初始密钥协商和后续密钥更新困难
- 应用: 主要应用于加密网络传递的数据, 例如分组、报文、文件、电子邮件内容等。



# 公钥数据加密

- 不对称密钥, 公钥和私钥分别用于加密和解密过程。
- 不仅可应用于数据加密, 还可以应用于数字签名。
- 私钥保密就能保证密文不被攻破
- 优点: 无需进行初始密钥的协商
- 缺点: 加密和解密算法的计算量较大, 计算成本较高
- 应用: 主要应用于密钥协商、数字签名。



# 传统数据加密概述

- 恺撒加密法
- 栅栏加密法和矩阵加密法
- 数据加密标准DES算法
- 三重DES算法
- 高级加密标准AES算法
- RC4算法
- 加密操作模式



# 凯撒加密法

- 恺撒加密法是将明文中的每个字母用该字母对应字母表的后续第3个字母替代，这里假定字母按照字母表顺序循环排列。
- 例如明文中的字母a对应密文中的字母D，明文中的字母x对应密文中字母A。采用凯撒加密法加密的举例如下：
- 明文： attack after dark
- 密文： DWWDFN DIWHU GDUN



# 通用凯撒加密算法

- W. Stallings将凯撒加密算法中的字母表移位数从3扩展到任意数 $k < 26$ , 这样, 就可以得到通用凯撒加密算法:

$$C = E(k, p) = (p + k) \text{ mode } 26$$

- 这样, 通用凯撒解密算法就可以表示为:

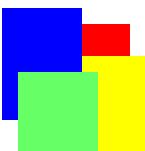
$$p = D(k, C) = (C - k) \text{ mod } 26$$

- 这里 $k$ 就是通用凯撒加密算法的密钥. 由于 $k$ 只有25个可能取值, 所以, 在已知加密/解密算法下, 只要尝试25种密钥, 就可以破译通用凯撒加密算法.
  - 如何计算通用凯撒加密法的密钥长度?



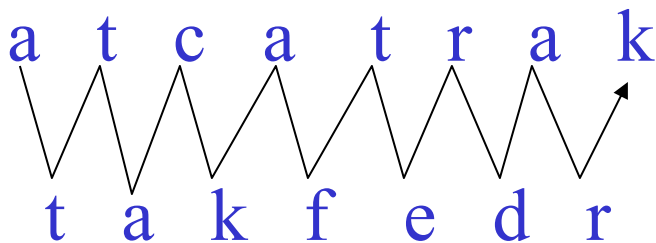
# 传统数据加密原理

- 传统数据加密包括两条原理: 替代和换位
- 替代: 将明文中的字母采用其它字母/数字/符号替代. 如果明文采用比特序列表示, 则将明文比特模式替代为其他比特模式.
  - 凯撒加密法就是采用替代原理设计的加密方法
- 换位: 将明文中的元素(字母/比特/字母组/比特组)进行某种形式的重新排列
  - 围栏加密方法、矩阵加密方法



# 围栏加密方法

- 最简单的一种换位加密算法是围栏方法, 将明文逐字符交替书写成为上下两行, 再逐行顺序读出上面一行和下面一行的字母序列
- 按照围栏方法对前面举例中的明文 “attack after dark” 的加密过程如下所示。
- 经过加密之后的密文就是一串没有意义的字符串: “ATCATRAKTAKFEDR”。





# 矩阵加密方法

- 围栏加密算法简单, 容易被破译。矩阵加密方法也是一种换位加密, 它将报文逐行写成一个 $n \times m$ 矩阵, 然后按照事先定义的列序列, 顺序读出相应列字母。
- 按列读出字母的列序号就构成了换位加密的密钥。

密钥: 25134

明文:

a t t a c

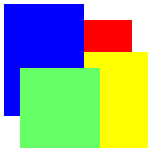
k a f t e

r d a r k

行

列

密文: TADCEKAKRTFAATR



# 数据加密标准DES

- 数据加密标准，英文缩写DES，是迄今为止应用最为广泛的标准化加密算法之一。
- DES是美国国家标准局(NBS，现在更名为美国国家标准与技术学会NIST)于1977年标准化的一种传统数据加密算法。
- DES是在国际商用机器(IBM)公司于1973年提出的一种加密方法的基础上，经过美国国家安全局(NSA)的验证和修改后标准化的加密方法。



# 数据加密标准DES特征

- IBM公司提交的加密算法是一种对称密钥的块加密算法，块长度为128个比特，密钥长度为128个比特。
- 该加密算法经过NSA(美国国家安全局)安全评估后，将其块长度更改为64个比特，密钥长度更改为56个比特，并且修改了原来加密算法中的替代矩阵S-盒。
- 对于NSA对DES算法的更改有以下评论：
  - (1) NSA降低密钥长度是为了降低DES加密算法的安全性，使得NSA在必要时，有能力破译DES加密系统。
  - (2) NSA修改原来加密算法中的S-盒，是为自己破译DES加密系统设置了后门。



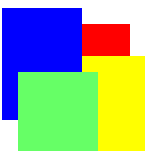
# DES算法概述

- DES加密算法是一种比较容易理解的加密算法。虽然其中具体的“替代”和“换位”的处理函数比较复杂，但这并不妨碍对DES加密算法的理解，也没有增加DES加密算法的计算复杂度。
- DES算法处理过程沿时间轴纵向可以分成前期处理、16次循环处理、后期处理部分，横向可以分成数据处理和密钥处理两个部分。



# DES算法的加密与解密

- DES加密处理分成三个部分:
  - (1) 64比特明文块通过初始排列IP(换位)处理;
  - (2) 通过16个轮回的替代和移位处理, 左右半换位形成预输出
  - (3) 预输出的64比特块进行逆初始排列 $IP^{-1}$ 处理, 产生64比特块密文
- DES解密过程与DES加密过程基本一样, 但是需要按照相反顺序使用子密钥, 即按照 $K_{16}, K_{15}, \dots, K_1$ 顺序处理每个轮回



## 三重DES算法

- 早在20世纪70年代颁布DES标准的时候，一些数据加密专家就预测到随着计算技术的发展，DES将难以满足安全性的需求，因此，提出了多重DES算法。
- 多重DES算法基本思路是：通过多次对数据块执行DES加密和解密操作，提高密文的安全性。
- 三重DES算法，简称为TDES、TDEA或者3DES，在1999年被接纳为NIST标准，该算法可以将DES的密钥扩展为112比特或者168比特长度。



## 三重DES算法(续)

TDES算法的思路十分简单：利用密钥 $K_1$ 对明文 $P$ 进行一次DES加密，利用密钥 $K_2$ 对密文 $C_1$ 进行一次DES解密，再利用密钥 $K_3$ 进行一次DES加密。

TDES加密算法的公式表示如下：

$$C = E(K_3, D(K_2, E(K_1, P)))$$

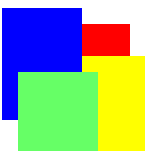
TDES解密过程正好与加密过程相反：

$$P = D(K_1, E(K_2, D(K_3, C)))$$



# DES算法分析

- DES算法最大弱点是密钥长度太短
- 虽然可以通过3DES算法可以将密钥扩展到112比特(当 $K_1=K_3$ )或者168比特(当 $K_1 \neq K_3$ )长度,但是,由于需要执行三次DES算法,其加密效率较低,无法满足NIST关于加密算法高效性的要求(目前不再是一类问题了! )。
- DES算法的另外一个缺陷是在软件中运行的效率较低。
- 为此NIST于1997开始在世界范围征集替代DES的加密算法,称为高级加密标准(AES)。



# RC4加密算法

- 与DES和AES加密算法不同，RC4是一种典型的流加密算法，它在1987年由公钥数据加密中著名的RSA加密算法的发明人之一R. Rivest提出。
- 由于RC4的简单性和开放源码，它被广泛应用于网络安全中。
- RC4流加密算法加密过程主要包括两个步骤：
  - (1) 利用密钥K生成一个伪随机比特序列
  - (2) 用该伪随机比特序列与明文进行“异或”运算产生密文。



# RC4加密算法的原理

- RC4流加密算法的解密过程也包括两个步骤：
  - 利用同样的密钥K生成相同的一个伪随机比特序列
  - 用该伪随机比特序列与密文进行“异或”运算得到明文。
- 在流加密算法中，由密钥K生成的伪随机比特序列称为“密钥流”，用KS(K)表示。
- RC4流加密算法可以用以下两个公式表示：

$$C = P \oplus KS(K)$$

$$P = C \oplus KS(K) = (P \oplus KS(K)) \oplus KS(K)$$

- 为了保证流加密算法的安全性，对于不同的明文，必须采用不同的KS
  - RC4流加密算法是传统加密算法吗？



# RC4加密算法的具体实现

- RC4加密算法分成两个部分：
  - 初始化部分，主要用于产生长度为256个八位位组的伪随机比特序列；
  - RC4处理部分，利用初始化产生的伪随机比特序列对明文（加密）或者密文（解密）进行逐个八位位组的“异或”操作；并且当对一个八位位组进行一次“异或”操作之后，就更改伪随机比特序列（类似于一次一密）。



# 加密操作模式

- 前面介绍的DES和AES加密算法都是块加密算法，它们只能对64比特或者128比特的数据进行加密。现实网络中传递的数据长度并不一定是64比特或者128比特？
- 问题1：如何在现实网络环境下针对任意长度的报文应用块加密算法？
- 问题2：是否可以使用块加密算法加密/解密数据流？



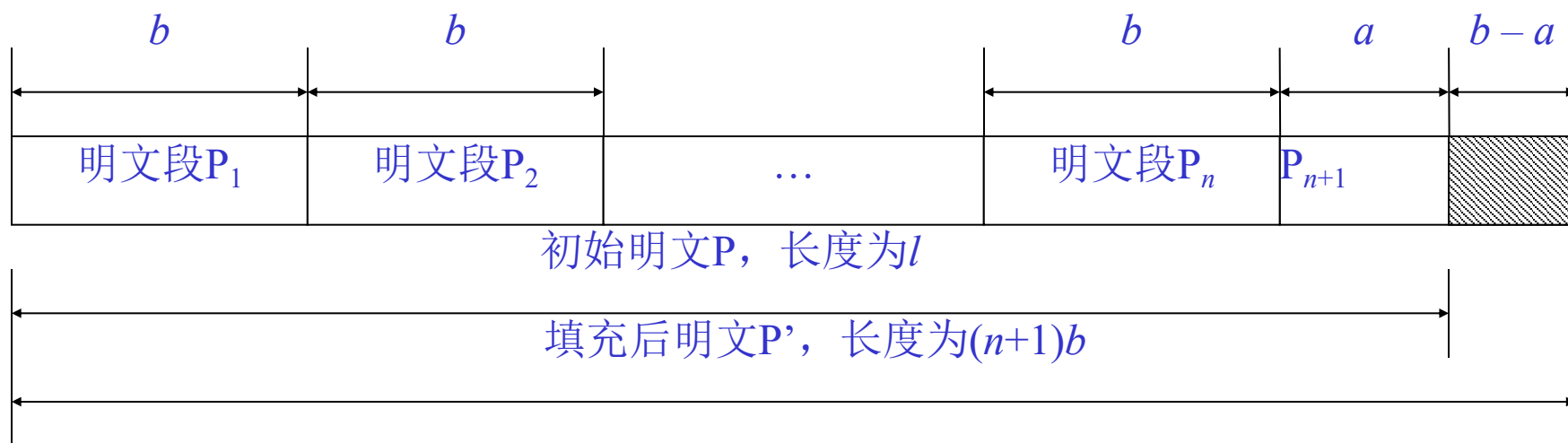
## 加密操作模式(续)

- 加密操作模式就是利用块加密算法对任意长度的数据块或者数据流进行加密或者解密操作的方式。
- NIST标准化了4种加密操作模式：
  - 电子密码簿(英文缩写ECB)模式
  - 密文块链接(英文缩写CBC)模式
  - 密文反馈(英文缩写CFB)模式
  - 输出反馈(英文缩写OFB)模式。



# 任意长度报文的处理

- 对于任意长度的报文可以采用分块和填充的方法，将报文转换成适合于块加密算法要求长度的数据块。

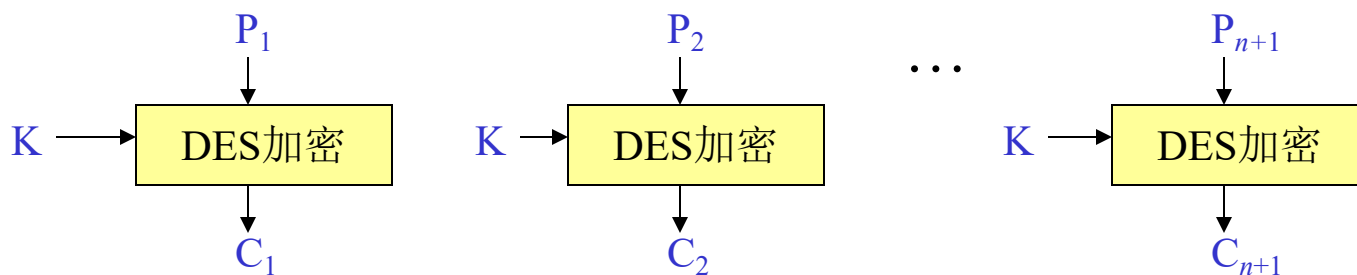


数据报文分段加密示意图



# 电子密码簿(ECB)模式

- 最简单的处理办法是电子密码簿ECB：对于一组长度都为 $b$ 的数据块，就可以采用相同密钥 $K$ 分别执行 $n+1$ 次加密算法 $E$ 。
- ECB的问题：对于给定某个密钥, 相同内容的数据块, 产生相同的密文块

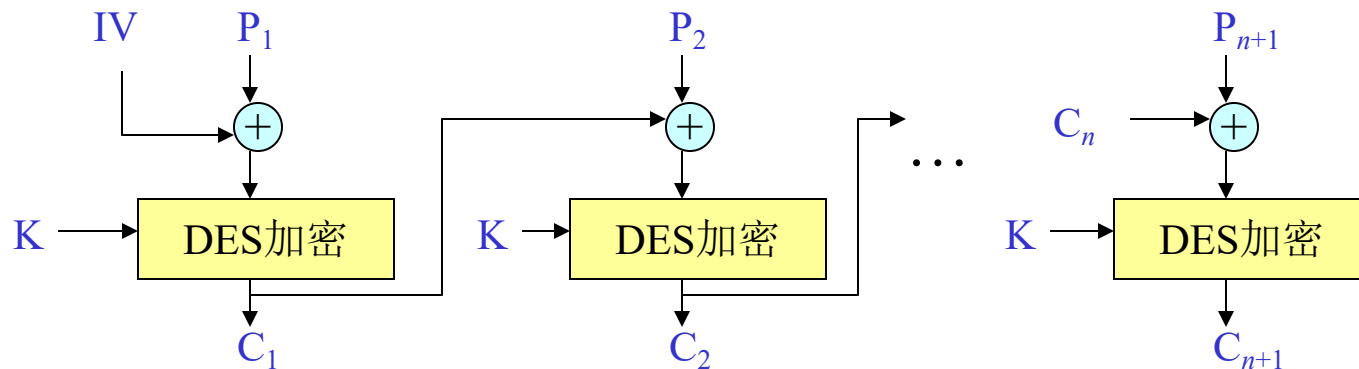


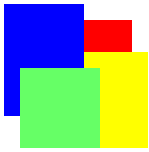
# 密文块链接(CBC)模式

- CBC模式中, 加密算法输入是前个密文块与当前明文块“异或”的结果, 每个块使用相同密钥
- CBC模式中加密和解密算法公式

$$C_j = E(K, P_j \oplus C_{j-1}), P_j = D(K, C_j) \oplus C_{j-1}, C_0 = IV$$

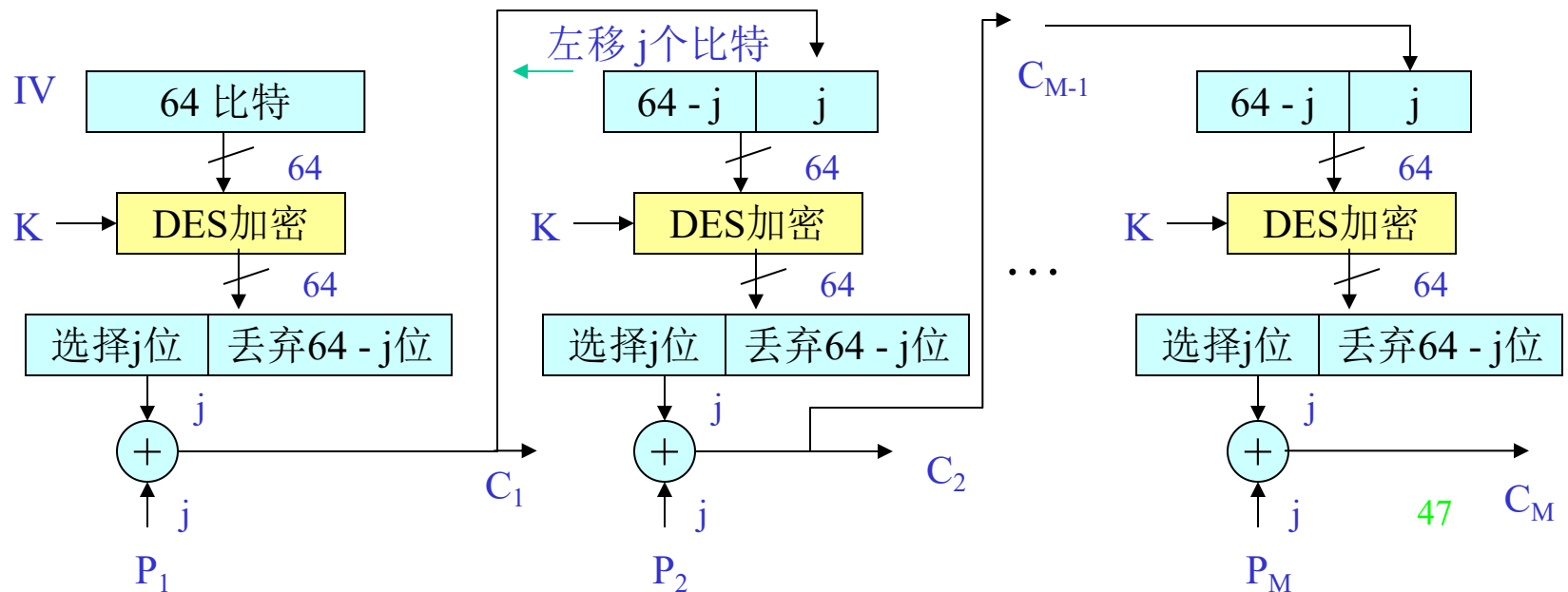
- CBC中与第一个明文块进行“异或”操作的初始化向量IV需要保密. 通常经过加密后传递





# 密文反馈(CFB)模式

- CFB可以将DES这类块加密算法转换成流加密算法. 流加密算法不需要填充数据, 无浪费.
- 假定传输单元长度 $j$ 比特( $j$ 通常为8), 与CBC类似, 其上次密文作为本次输入.





## 密文反馈(CFB)模式公式表示

- 与CBC不同, CFB仅仅对初始值与反馈值的合并数据加密后, 再与明文“异或”得到密文。以上加密/解密过程可以采用公式表示如下:

$$R_j = (R_{j-1} * 2^l \bmod 2^b) + C_{j-1}$$

$$C_j = S(l, E(K, R_j)) \oplus P_j$$

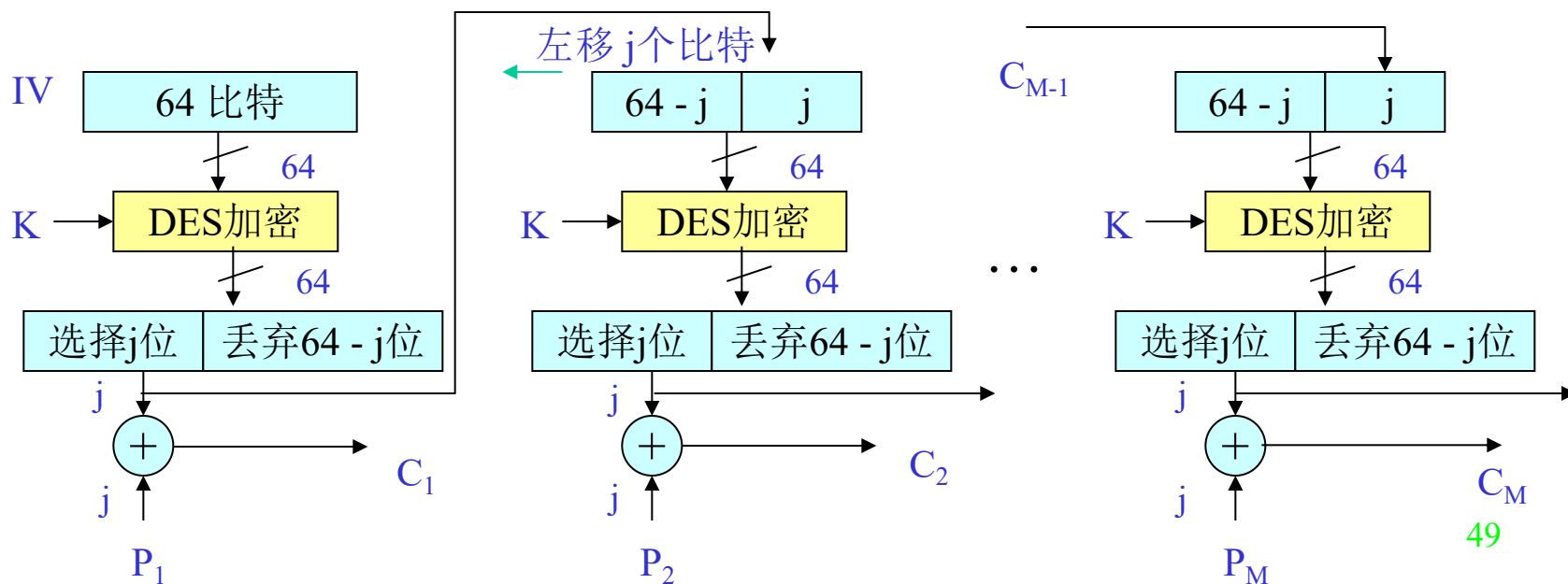
$$P_j = S(l, E(K, R_j)) \oplus C_j$$

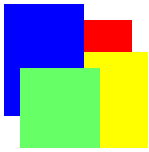
$$R_0 = IV$$



# 输出反馈(OFB)模式

- 输出反馈(OFB)加密操作模式也是一种流加密操作模式。与CFB模式类似，但却利用块加密算法输出的反馈生成随机比特序列(流密钥)，然后，利用该流密钥与明文数据流进行“异或”运算，得到密文数据流。





# 公钥数据加密概述

- 公钥数据加密发展动因
- 公钥数据加密基本原理
- RSA公钥加密算法
- Diffie-Hellman密钥生成算法
- 公钥数据加密与密钥管理



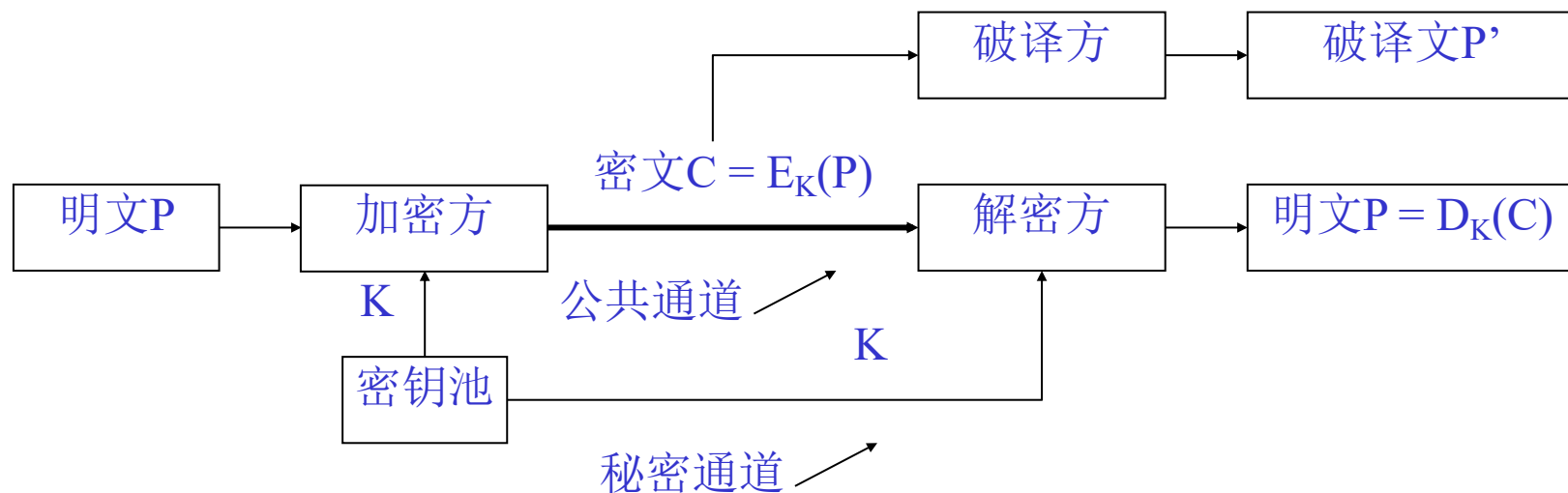
# 公钥数据加密发展动因

- 公钥数据加密发展动因来源于电信网(公共网络)环境下安全数据传递的应用需求。
- 在电信网环境下，数据传递存在以下两类安全威胁：
  - 其一是窃取电信网上传递的数据(破坏保密性)；
  - 其二在电信网传递的数据中插入虚假的数据(破坏完整性)。
- 为了解决第一个问题，可以采用数据加密方法对电信网传递的数据进行加密，但需要事先协商密钥。
- 如何解决在收发双方互不信任时的数据完整性问题？这是传统的数据加密难以有效解决的问题。



# 传统加密算法的数据问题\*

- 在通信网环境下，传统数据加密需要单独秘密通道(例如短信)实现快速的密钥分发，传统的数据加密可以解决电信网环境的数据保密问题。由于收发双方采用相同密钥，无法解决收发双方在数据完整性的纠纷问题。



对称密钥加密算法原理示意图



# 公网数据加密的解决方案\*

- 为了解决在电信网环境下任意两个互不相识的用户之间能够进行安全数据传递问题，M. Diffie和W. Hellman设计了两个方案：

方案1：采用公钥数据加密系统，将传统数据加密的密钥分解成加密密钥PK和解密密钥VK，从PK无法推导出VK，这样，就可以公开加密密钥PK，由接收方保管自己的解密密钥VK。这就需要采用公钥加密算法。

方案2：采用“公钥分发系统”，通过公开交互的信息，可以生成只有通信双方才知道的密钥。再采用传统数据加密进行加密和解密处理。这就需要采用Diffie-Hellman密钥协商算法。



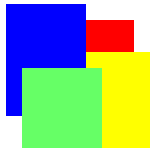
# 公网数据完整性解决方案\*

- 为了解决电信网环境下数据完整传递的问题，必须设计一种机制，使得该发送方传递的数据，除发送方之外，任何其他一方都无法修改数据。这样，才能通过电信网传递商业合同。
- 由于传统数据加密的加密算法中发送方和接收方共用同一个密钥，则接收方接收后可以更改数据。这样，传统数据加密无法解决电信网环境下数据完整传递的问题。
- 公钥数据加密中只有一方掌握私钥VK，如果发送方采用私钥加密报文摘要后传递（数字签名），则其他一方难以做到修改报文而不被发觉。公钥数据加密可解决数据完整传递问题。



# 公钥数据加密系统定义\*

- W. Diffie和M. Hellman于1976年首先给出了公钥数据加密系统的定义：
- 一个公钥数据加密系统由一对加密算法E和解密算法D构成，该公钥数据加密系统采用一个密钥对集合 $KS = \{(PK, VK)\}$ ，对于任何一个KS集合中的密钥对(PK, VK)和任何一个明文P，存在以下特性：
  - (1) 采用密钥对(PK, VK)中任何一个密钥针对明文P执行加密算法E，都可以采用另一个密钥针对密文进行解密。——对等性（加密与解密）



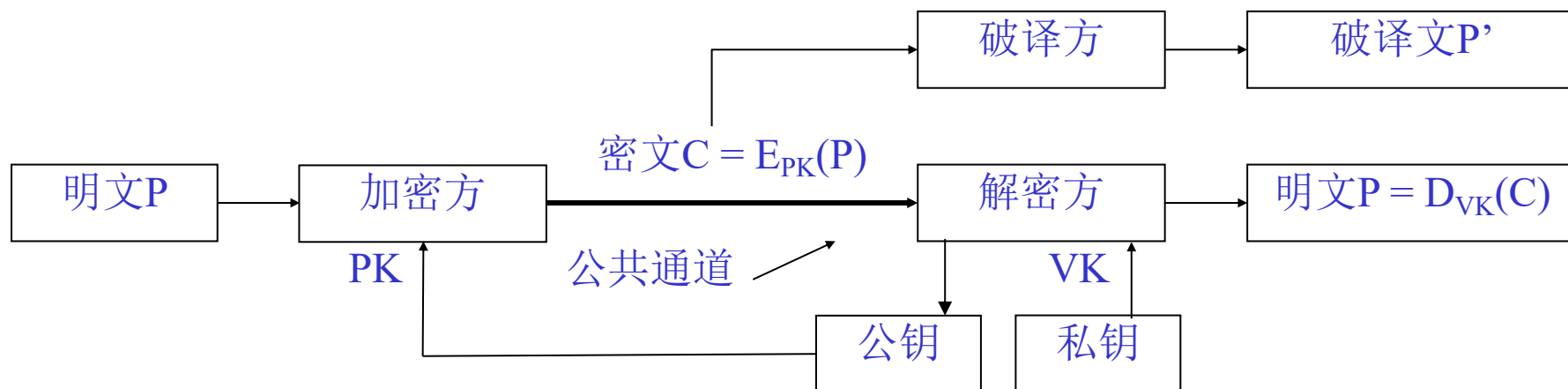
## 公钥数据加密系统定义(续1)\*

- (2) 对于掌握了密钥对(PK, VK), 则加密算法E和解密算法D都是容易计算的。——可用性(加密+解密)
  - (3) 如果公开密钥对中的一个密钥, 例如PK, 则无法通过计算推导出另一个密钥, 例如VK。——安全性(解密)
  - (4) 如果只掌握了密钥对中的一个密钥PK, 并且利用该密钥将明文P加密得到密文C, 则无法再利用该密钥将C进行解密得到明文P。——唯一性(解密)
- 以上4个特性较为完整地刻画了公钥数据加密系统的特征。



# 公钥数据加密系统示意图\*

- 公钥数据加密系统中的加密算法采用了不同的加密密钥和解密密钥，所以，也称为不对称密钥加密算法。其原理如下图所示。

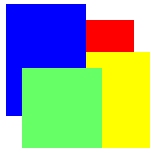


公钥加密算法原理示意图



# 公钥数据加密系统定义\*

- W. Diffie和M. Hellman于1976年首先给出了公钥数据加密系统的定义：
- 一个公钥数据加密系统由一对加密算法E和解密算法D构成，该公钥数据加密系统采用一个密钥对集合 $KS = \{(PK, VK)\}$ ，对于任何一个KS集合中的密钥对(PK, VK)和任何一个明文P，存在以下特性：
  - (1) 采用密钥对(PK, VK)中任何一个密钥针对明文P执行加密算法E，都可以采用另一个密钥针对密文进行解密。——对等性（加密与解密）



## 公钥数据加密系统定义(续1)\*

- (2) 对于掌握了密钥对(PK, VK), 则加密算法E和解密算法D都是容易计算的。——可用性 (加密+解密)
  - (3) 如果公开密钥对中的一个密钥, 例如PK, 则无法通过计算推导出另一个密钥, 例如VK。——安全性 (解密)
  - (4) 如果只掌握了密钥对中的一个密钥PK, 并且利用该密钥将明文P加密得到密文C, 则无法再利用该密钥将C进行解密得到明文P。——唯一性 (解密)
- 以上4个特性较为完整地刻画了公钥数据加密系统的特征。



# RSA公钥加密算法

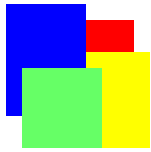
- RSA公钥加密算法是基于数论中的欧拉定理和费马定理设计的一种加密算法，其安全性主要是基于“大数分解”的不可解特性。
- RSA公钥加密算法可以分成两个部分：
  - RSA公钥加密算法的加密和解密过程；
  - RSA公钥加密算法的“密钥对”选择和生成过程。



# RSA算法加密/解密过程

- 为了利用一个公钥( $e, n$ )对一个报文 $M$ 进行加密，这里 $e$ 和 $n$ 是一对正整数，可以采用以下过程：
  - (1) 将报文 $M$ 表示成一个0到 $n - 1$ 的整数。如果 $M$ 较长，可以将 $M$ 分解成多个数据块，分别进行多次加密。
  - (2) 将 $M$ 进行 $e$ 次乘法运算，然后对乘积取 $n$ 的模，这样，就得到 $M$ 的密文 $C$ 。

$$C = E(M) = M^e \pmod{n}$$



## RSA算法加密/解密过程(续)

(3) 如果需要对密文C进行解密，则只需要对C进行 $d$ 次乘法运算，然后再对乘积取 $n$ 的模，这样，就得到报文M。

$$M = D(C) = C^d \pmod{n}$$

- 这里 $(d, n)$ 就是与公钥 $(e, n)$ 对应的私钥，这是需要该“密钥对”所有者秘密保存的密钥。
- 这里的 $e$ ， $d$ 和 $n$ 是需要用户在生成“密钥对”过程中选择和生成的正整数。

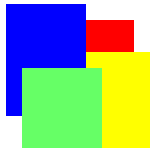


# RSA算法的密钥选择

- 为了使用RSA加密算法，首先需要按照以下方法选择并生成RSA公钥加密算法的“密钥对”。
  - (1) 选择两个很大的“随机”素数 $p$ 和 $q$ ，这两个素数的乘积就是RSA密钥中的正整数 $n$ ，即

$$n = p * q$$

- 如果 $p$ 和 $q$ 足够大，即使公开 $n$ ，则根据目前的计算能力，也无法分解出 $p$ 和 $q$ 。



## RSA算法的密钥选择(续)

(2) 选择一个很大的随机整数 $d$ ，使得该整数与 $(p - 1) * (q - 1)$ 的最大公因子为1

(3) 从 $p$ ， $q$ 和 $d$ 中计算出 $e$ ， $e$ 是以 $(p - 1) * (q - 1)$ 为模的 $d$ 的倒数，即

$$e * d = 1 \pmod{(p - 1) * (q - 1)} \text{ 即:}$$

$$e * d - i * (p - 1) * (q - 1) = 1$$

- 私钥是根据一定规则选择的，而公钥是计算得出的。



## RSA算法举例 (1-1)

例题1：选择 $p = 47$ ,  $q = 59$ ,  $n = 47 * 59 = 2773$ ,  
 $d = 157$ 。  $\varphi(n) = 46 * 58 = 2668$ , 利用欧几里德  
算法计算 $e$ 如下：

$$2668 = 157 \times 16 + 156; \quad 157 = 156 \times 1 + 1$$

$$156 = 2668 - 157 \times 16$$

$$\rightarrow 157 = (2668 - 157 \times 16) \times 1 + 1$$

$$\rightarrow 157 \times 17 - 2668 \times 1 = 1$$

对照：  $e * d = 1 \pmod{(p-1) * (q-1)}$

这样，可知 $e = 17$

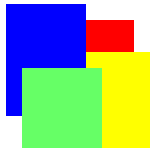


## RSA算法举例 (1-2)

- 以下对 “ITS ALL GREEK TO ME” 进行加密。首先采用00表示空格、01表示A、26表示Z，对该句子进行编码，得到以下数据：

0920 1900 0112 1200 0718 0505 1100 2015  
0013 0500

- 以两个字符（字母或空格）为一个加密数据块，则该数据块最大取值 $2626 < 2773 = n$ ，可以采用RSA加密算法。



## RSA算法举例(1-3)

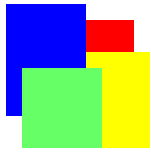
- 以下仅对第一个数据块进行加密：

$$920^{17} = 948 \pmod{2773}$$

- 以上整个英文句子加密后的密文为：

0948 2342 1084 1444 2663 2390 0778 0774 0219 1655

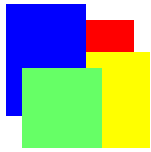
- 对第一个数据块的解密算式可以按照加密算式进行，其结果为： $948^{157} = 920 \pmod{2773}$ 。



## RSA加密算法举例(2-1)\*

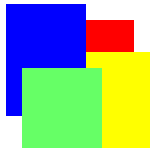
例题2：假定 $p=79$ ， $q=67$ ， $n = 79 \times 67 = 5293$ ， $\varphi(n) = 78 \times 66 = 5148$ ， $d=127$ ，求与 $d$ 对应的公钥 $e$ ，并验证公钥的正确性

- $5148 = 40 \times 127 + 68$ ， $127 = 1 \times 68 + 59$ ， $68 = 1 \times 59 + 9$ ， $59 = 6 \times 9 + 5$ ， $9 = 1 \times 5 + 4$ ， $5 = 1 \times 4 + 1$
- $5148 - 40 \times 127 = 68$ ，
- $127 - 1 \times 68 = 59 \rightarrow 127 - (5148 - 40 \times 127) = 59 \rightarrow 41 \times 127 - 5148 = 59$



## RSA加密算法举例(2-2)\*

- $5148 - 40 \times 127 = 1 \times (41 \times 127 - 5148) + 9 \rightarrow$   
 $2 \times 5148 - 81 \times 127 = 9$
- $41 \times 127 - 5148 = 6 \times (2 \times 5148 - 81 \times 127) + 5$   
 $\rightarrow 527 \times 127 - 13 \times 5148 = 5$
- $2 \times 5148 - 81 \times 127 = 527 \times 127 - 13 \times 5148 + 4$   
 $\rightarrow 15 \times 5148 - 608 \times 127 = 4$
- $527 \times 127 - 13 \times 5148 = 15 \times 5148 - 608 \times 127 + 1 \rightarrow$   
 $1135 \times 127 - 28 \times 5148 = 1 \rightarrow e = 1135$
- RSA密钥的验证:  $(0321)^{1135} \pmod{5293} = 0072,$   
 $(0072)^{127} \pmod{5293} = 0321$



## RSA加密算法举例(3-1)

例题3:  $p=53$ ,  $q=67$ ,  $n=53 \times 67=3551$ ,  
 $\varphi(n) = 52 \times 66=3432$

如果选择私钥 $d=131$ , 求对应的公钥 $e$

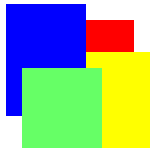
$$3432 = 26 \times 131 + 26$$

$$131 = 5 \times 26 + 1$$

$$131 = 5 \times (3432 - 26 \times 131) + 1 \rightarrow$$

$$131 \times 131 - 5 \times 3432 = 1 \rightarrow e = 131$$

• 这个公钥是否合理? 为什么?

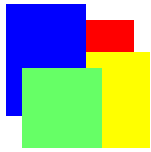


## RSA加密算法举例(3-2)

例题3（续）：已知 $p=53$ ， $q=67$ ， $n=53 \times 67 = 3551$ ， $\varphi(n) = 52 \times 66 = 3432$

如果选择私钥 $d=137$ ，求与该 $d$ 对应的公钥 $e$

- $3432 = 25 \times 137 + 7$
- $137 = 19 \times 7 + 4$
- $7 = 1 \times 4 + 3$
- $4 = 1 \times 3 + 1$
- $3432 - 25 \times 137 = 7$



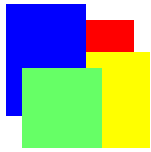
## RSA加密算法举例(3-3)

- $137 = 19 \times (3432 - 25 \times 137) + 4 \rightarrow 476 \times 137 - 19 \times 3432 = 4$
- $3432 - 25 \times 137 = 476 \times 137 - 19 \times 3432 + 3 \rightarrow 20 \times 3432 - 501 \times 137 = 3$
- $476 \times 137 - 19 \times 3432 = 20 \times 3432 - 501 \times 137 + 1 \rightarrow 977 \times 137 - 39 \times 3432 = 1 \rightarrow e = 977$
- 密钥的验证:  $(0920)^{977} \pmod{3551} = 0088$ ,  
 $(0088)^{137} \pmod{3551} = 0920$



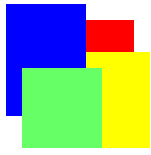
# RSA算法分析\*

- R. L. Rivest等人的分析：攻破RSA加密算法的计算复杂度等同于分解大数 $n$ 的计算复杂度。
- 假定 $n = 2^l$ ，则当 $l > 336$ （相当于100个十进制位），时间复杂度可以小于 $O(2^{l/8})$  ( $< O(2^{50})$ )的分解大数 $n$ 的算法。
- 1994年, 一个小组利用互联网上1600台计算机, 经过8个月的计算, 攻破了公钥长度为129位十进制数(约428比特)的RSA(这种攻击意义不大).
- 现在通常认为, 采用1024比特密钥长度(约300位十进制数)是比较安全的.



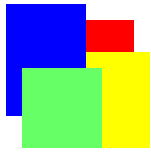
# Diffie-Hellman密钥生成算法

- 这里介绍的Diffie-Hellman密钥生成算法，实际上并不是W. Diffie和M. Hellman提出的公钥数据加密中的公钥加密算法，而只是基于公钥数据加密的密钥协商算法。
- 而且Diffie-Hellman算法生成的密钥也并不是公钥数据加密中使用的密钥，而是在传统数据加密中使用的密钥。



# Diffie-Hellman算法的加密过程\*

- 如果在电信网上的一方A试图与电信网上的另外一方B进行通信，则
  - A首先通过“电话黄页簿”获取B公布的公钥以及相关的参数；
  - 然后，利用自己的私钥对B的公钥进行指数运算后再取模，得到密钥 $K_{A,B}$ ；
  - 最后，A利用 $K_{A,B}$ 作为密钥，利用传统加密算法（例如DES算法）加密数据后传递给B。



# Diffie-Hellman算法的解密过程\*

- 在电信网上的另一方B收到了A发送来的加密报文之后，则
  - 首先通过“电话黄页簿”获得A公布的公钥以及相关的参数；
  - 然后，利用自己的私钥对A的公钥进行指数运算后再用相同的数取模，得到密钥 $K_{B,A}$ ；
  - Diffie-Hellman算法可以保证 $K_{B,A} = K_{A,B}$ ，这样，B就可以利用 $K_{B,A}$ 解密A采用传统加密算法和密钥 $K_{A,B}$ 生成的密文。



# Diffie-Hellman算法基本过程\*

- 假设 $q$ 是一个素数， $\alpha$ 是 $(1, q)$ 范围中的一个素数，利用Diffie-Hellman密钥生成算法生成共享密钥过程如下：
  - (i) A方从 $\{1, 2, \dots, q-1\}$ 中选择一个随机整数 $X_A$ 作为保密字保存好，将 $Y_A = \alpha^{X_A} \bmod q$ 计算值连同A的名字、地址、 $\alpha$ 和 $q$ 值等信息放置在公共文件中。
  - (ii) B方从 $\{1, 2, \dots, q-1\}$ 中选择一个随机整数 $X_B$ 作为保密字保存好，将 $Y_B = \alpha^{X_B} \bmod q$ 计算值连同B的名字、地址、 $\alpha$ 和 $q$ 值等信息也放置在公共文件中。——注意：公开的数据都是计算得出的数据，并且难以反推出选择的、需要保密的数据。



## Diffie-Hellman算法基本过程(续)\*

(iii) 如果A要与B进行保密通信，则A从公共发布中取出B公开的数值 $Y_B$ ，利用自己保存的保密字对 $Y_B$ 进行指数运算，得到密钥

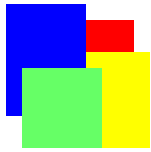
$$K_{A,B} = (Y_B^{X_A}) \bmod q$$

(iv) 同样B也可以从公共发布中取出A公开的数值 $Y_A$ ，利用自己保存的保密字对 $Y_A$ 进行指数运算，得到密钥

$$K_{B,A} = (Y_A^{X_B}) \bmod q$$

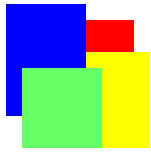
而  $K_{B,A} = (Y_A^{X_B}) \bmod q = \alpha^{X_A \cdot X_B} \bmod q = \alpha^{X_B \cdot X_A} \bmod q = K_{A,B}$ 。

(v) 随后A与B就可以利用双方共享的 $K_{A,B}$ 进行数据传统加密和解密的操作。



# Diffie-Hellman算法举例

- 对于Diffie-Hellman密钥生成算法，假定  $q=71$ ， $\alpha=53$ ， $X_A=21$ ， $X_B=17$ ，则
- $Y_A = 53^{21} \pmod{71} = 66$
- $Y_B = 53^{17} \pmod{71} = 69$
- $K_{A,B} = 69^{21} \pmod{71} = 46$
- $K_{B,A} = 66^{17} \pmod{71} = 46 = K_{A,B}$



# Diffie-Hellman算法安全性分析

- Diffie-Hellman算法生成的密钥在选择合适的 $q$ 值的条件下是安全。
- 因为计算共享密钥 $K_{A,B}$ 最多需要花费 $2\log_2 q$ 次运算，
- 攻击者C试图利用 $Y_A$ 或者 $Y_B$ 破译 $K_{A,B}$ 至少需要 $q^{1/2}$ 次运算。
- 例如假定 $q$ 取值为略小于 $2^b$ 的一个素数，则A和B计算共享密钥需要花费 $2b$ 次运算，而破译 $K_{A,B}$ 至少需要花费 $2^{b/2}$ 次运算。
- 如果 $b$ 取值为200(即200个比特长度)，则A和B计算共享密钥只花费400次运算，而破译该密钥需要花费 $2^{100}$ 次运算，相当于 $10^{30}$ 次运算。



# 公钥数据加密与密钥管理

- 公钥数据加密中的公钥并不能随意公开，而是必须通过权威机构公开而权威地发布。因为公钥在某种程度上对应了某人的身份。
- 虽然按照公钥数据加密的规则，已知公钥并不能够破解对应的私钥。但网络攻击者可以通过发布某些用户的虚假公钥，假冒这些用户，利用公钥数据加密系统与其他用户进行保密通信，获取通信对方的保密数据。



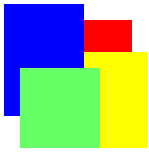
# 公钥数据加密与密钥管理(续)

- 像最初设想的那样，通过电话黄页或者公共信息服务器公布的公钥是无法防范网络攻击者的，设置“只读”权限的共用文件根本无法防范网络攻击。公钥管理应该成为整个系统安全管理的一个重要环节。
- 目前公钥都通过公钥基础设施（PKI）进行管理和发布。它是实现电子商务的重要基础设施。
- PKI将在第3章“真实性验证技术”一节中介绍。



# 真实性验证技术与应用

- 真实性验证基本概念
- 报文真实性验证
- 身份真实性验证协议
- 公钥基础设施PKI与真实性验证



# 真实性验证基本概念

- 真实性验证的作用
- 真实性验证技术的分类
- 真实性验证的内容
- 真实性验证的方式



# 真实性验证技术分类

- 网络安全中的真实性验证技术可分成两大类：身份真实性验证和报文真实性验证
- 身份真实性验证主要是识别网络实体的身份真实性，人(或信任主体)可以参与该验证过程。
  - 计算机安全系统中的真实性验证技术通常采用登录账户、以及密码或者密码+指纹的身份真实性验证技术。
  - 网络安全系统中的身份真实性验证需要输入验证码，为什么？



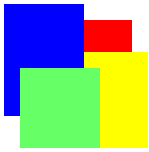
## 真实性验证技术分类(续)

- 报文真实性验证主要识别接收到的报文真实性，验证过程中人无法直接参与。
  - 网络安全系统中报文传递系统的数据真实性验证属于报文真实性验证。
  - 报文真实性验证不涉及人机交互, 这是通过程序设置的真实性验证过程。
- 互联网环境下的身份真实性验证可以作为网络访问控制中的第一个控制环节；
- 互联网环境下的报文真实性验证可以作为电子商务中防范合同欺诈（伪造或篡改合同）的技术手段。



# 身份真实性验证方法\*

- 人类的身份真实性验证方法可以根据具体验证的内容不同分成：
  - 基于知识的真实性验证（密码登录的账户），
  - 基于标志的真实性验证（采用员工卡的门禁），
  - 以及基于特征的真实性验证（采用指纹的门禁）。
- 这样，真实性验证的内容可以包括：被验证者掌握的“知识”，被验证者拥有的“标志”，以及被验证者唯一具有的“特征”。
  - 具有“人工智能”装置的身份真实性可以借鉴这种验证方法



# 真实性验证内容通俗说法

- 真实性验证的内容, 可以采用以下三句通俗的说法:
  - What do you know? 例如用户账户管理系统
  - What do you have? 例如标识卡系统
  - What are you? 例如指纹识别系统



# 基于知识的真实性验证\*

- 基于知识的真实性验证，也就是根据被验证者“知道什么”确定其真伪。例如计算机安全系统中的用户账户管理系统，就是一种基于知识的真实性验证技术。这种真实性验证最容易掌握，也最容易被假冒。
- 在身份真实性验证技术中，基于知识的真实性验证机制主要是指用户登录用户账户管理系统；
- 在报文真实性验证技术中，基于知识的真实性验证机制主要指采用不可伪造的、事先约定的保密字或密钥的报文验证技术。



# 基于标志的真实性验证\*

- 基于标志的真实性验证，也就是根据被验证者“拥有什么”确定其真伪。只要被验证者拥有具有“标志”意义的物品，例如银行发行的信用卡，或者登录计算机系统的智能卡等。
- 在身份真实性验证技术中，基于标志的真实性验证机制主要指采用身份识别卡的计算机或者网络登录系统。
- 在报文真实性验证技术，基于标志的真实性验证机制主要指采用加密标识的报文验证技术。



# 基于特征的真实验证\*

- 基于特征的真实验证，也就是根据被验证者“是什么”确定其真伪。这种被验证者的“特征”通常是指不可假冒的、可以唯一标识被验证者的特征。
- 在身份真实性验证技术中，基于特征的真实验证机制主要指人体眼虹验证系统，以及指纹验证系统。
- 在报文真实性验证技术中，基于特征的真实验证机制主要是指基于包含了不可加密标志的“报文摘要(报文哈希值、报文指纹)”的报文真实性技术。
  - 注：由于报文可以被修改，所以只是报文摘要无法验证其真伪，必须采用报文验证码进行报文真实性验证。



# 真实性验证内容的组合

- 以上介绍了3中真实性验证的内容也可以简称为：“所知(对应英文What you know)”、“所有(What you have)”和“所是(What you are)”。
- 由于单项内容难以保证真实性验证的安全性，通常采用多项内容组合的方法进行真实性验证。
  - 例如采用“所知” + “所有”进行真实性验证，持有身份标识卡的用户不仅需要插入标识卡，还需要输入口令。
  - 例如采用“所有” + “所是”进行真实性验证，用户不仅需要识别指纹，还需要插入标识卡(报文的保密字)。



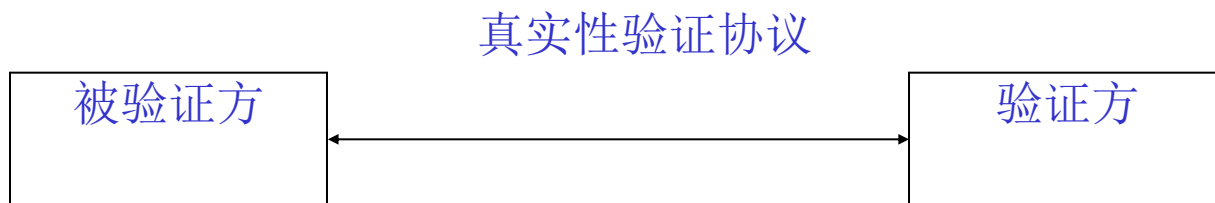
# 真实性验证的交互方式

- 在网络安全中，根据真实性验证参与方的数目，真实性验证可以分成**双方交互方式**、**三方交互方式**、或者**多方交互方式**(区块链的标识管理技术)。
- **双方交互方式**
  - 双方交互方式是指在真实性验证过程中，只涉及两个网络实体：真实性验证方和被验证方，双方通过交互真实性验证协议，**单向**或者**双向**验证身份。



# 双方交互的方式

- 单向真实性验证指只有真实性验证方验证对方身份真伪；
- 双向真实性验证指真实性验证方和被验证方相互进行真实性验证。



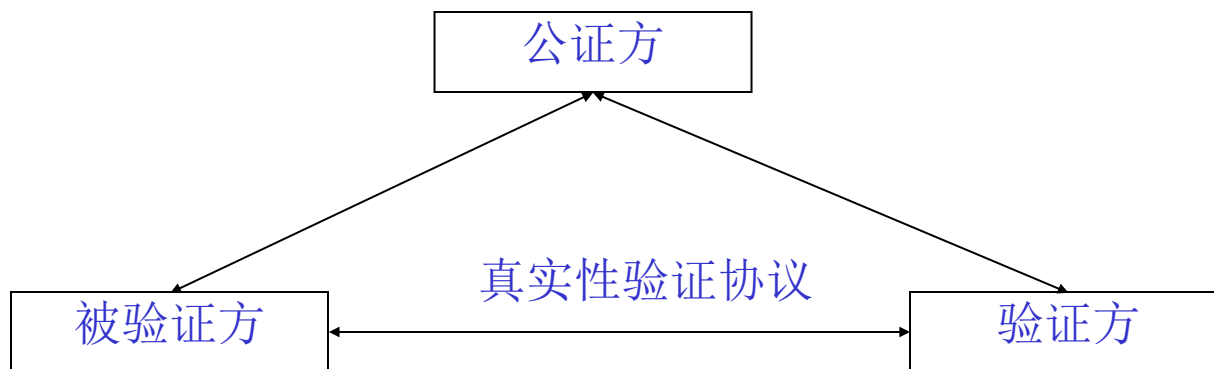
双方真实性验证方式



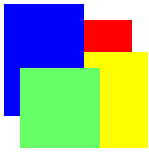
# 三方交互方式

- 三方交互方式

- 交互双方需要通过作为**公证方**的第三方, 才能相互验证身份的真实性
- 交互双方**不处于相同的信任域**, 需要通过第三方公证才能建立彼此信任.



三方交互方式



# 报文真实性验证的方法

- 报文真实性验证没有交互过程，主要采用方法：
- 报文摘要，提取报文的数据特征的方法，主要通过安全哈希函数，计算整个报文的哈希值(报文摘要)。如果该报文任何内容被修改，则哈希值将更改。
- 报文验证码：包括了报文发送方身份真实特征信息(保密字)的报文摘要，就成为报文验证码。
- 数字签名：采用报文发送方的私钥加密后报文摘要，就是该报文发送方的数字签名。



## 三方参与的报文真实性仲裁\*

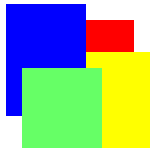
- 电子商务应用中, 通常采用三方参与的报文真实性仲裁方式
- 数字签名技术, 提供了三方参与的报文真实性仲裁的支撑。报文接收方采用权威发布的公钥解密报文摘要, 验证数字签名的真实性。
  - 一旦出现双方争执, 接收方可以将接收的报文提交给第三方进行验证, 确定是否是发送方发送的报文——不可抵赖性。



# 报文真实性验证

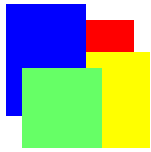
---

- 报文真实性验证基本概念
- 报文摘要算法MD5
- 安全哈希算法SHA-1
- 哈希函数的报文验证码算法HMAC
- 生日现象与生日攻击
- 数字签名



# 报文真实性验证基本概念

- 报文真实性验证也称为“报文验证”，它的目标是验证报文发送方的真实性，以及报文在传递过程中的完整性。
  - 发送方的真实性是指发送方真正具有发送方标识
  - 报文的完整性是指报文在传递过程中，除了正常协议处理而在报文中产生的改动外，报文的任何部分都没有被随意修改。
- 报文验证并不能验证报文到达的及时性以及报文到达的有序性。这些特性需要利用身份真实性验证协议加以验证。



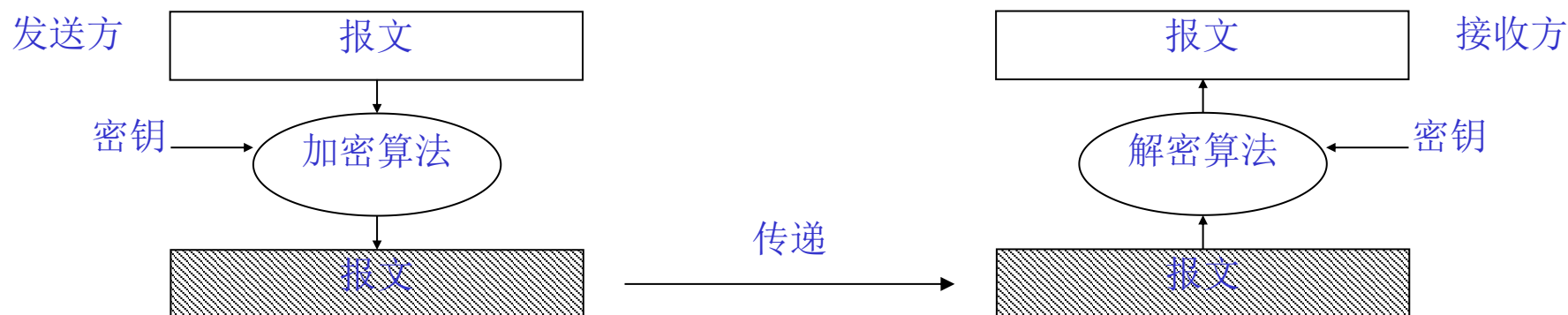
# 采用加密的报文验证方法

- 报文验证方法可以采用两种：一种是加密报文验证方法，另一种是非加密报文验证方法。
- 采用加密的报文验证方法具体可以分成以下3种方法：
  - 加密整个报文的报文验证方法
  - 加密报文校验和的报文验证方法
  - 附加加密文块的报文验证方法



# 加密整个报文的方法

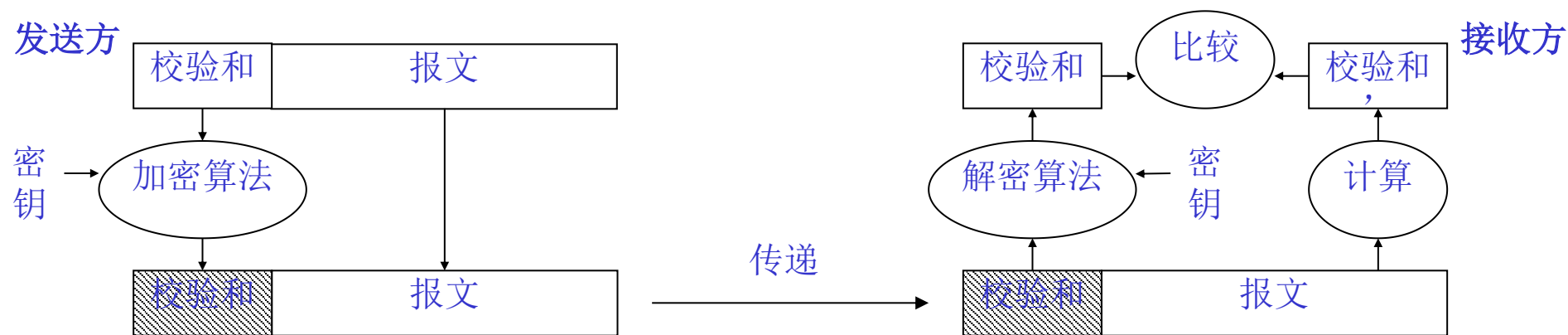
- 报文验证的最简单方法就是采用加密方法：可以**直接对整个报文进行加密**后再发送，接收方接收到之后首先解密，然后再接收报文。
  - 这类方法的报文真实性**依赖于**双方约定的**密钥的保密性**
- 这种方法的**处理效率较低**，因为**加密和解密过程都是消耗**计算资源较多的操作。

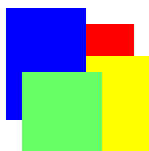




# 加密报文校验和的方法

- 报文验证也可以仅仅加密报文的校验和，而不加密整个报文。一般报文都具有校验和，用于检测在报文传递过程中是否出现差错。如下图所示。
  - 由于校验和容易被假冒，这类方法存在被假冒的风险





## 加密报文校验和的方法(续)

- 一般报文中的校验和都是采用奇偶校验方法生成的，它只能检测出在校验和同一个二进制位置上出现的奇数次改动。
- 根据校验和的特征可知，校验和并不可靠。网络攻击者可以同时修改偶数个比特，使得修改后报文的校验和与修改前报文的校验和相同，



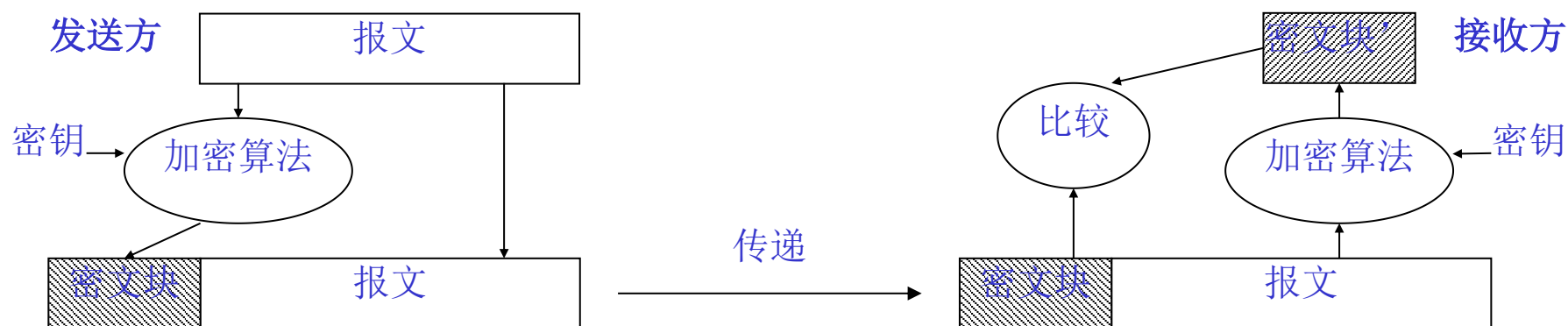
## 附加密文块的方法

- 采用传统的加密方法以及加密操作模式，例如采用DES+CBC加密方式，取密文的最后一个密文块，附加在报文后面传递给接收方。
- 接收方收到报文后，同样对接收到的报文进行加密，得到最后一个密文块。然后，比较接收到的密文块与计算得到的密文块是否相同，如果相同，则报文在传递过程中没有被修改。否则，报文在传递过程中被修改。

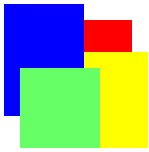


## 附加密文块的方法(续)

- 这种报文验证方法在发送方和接收方都需要对  
整个报文执行加密算法，而加密算法是一种计  
算开销较大的操作，故开销仍然比较大。

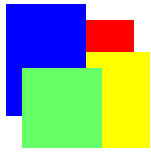


附加密文块的报文验证方法



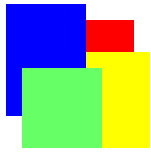
# 报文摘要与密码哈希函数

- 加密校验和的报文验证方法以及附加加密文块的报文验证方法揭示了报文验证的一个关键问题：
  - 如何能够采用较为简便的算法，计算出一个固定长度的、能够反映报文特征的数据块？
- 这个固定长度的、能够反映报文特征的数据块就称为报文摘要。
- 目前通常采用密码哈希函数生成报文摘要
- 这种报文摘要算法不同于加密算法。



# 报文摘要与密码哈希函数(续1)

- 假定这种哈希函数为 $H$ ，报文为 $m$ ，哈希值为 $h$ ，即 $h = H(m)$ 。这个哈希函数必须具备以下几个特征。
- **不可逆性**：如果 $h = H(m)$ ，并且已知 $h$ ，则在现有的计算条件下无法推导出 $m$ 。
  - 即无法从哈希值中推导出报文内容。
- **不可替代性**：如果 $h = H(m)$ ，并且已知 $h$ ，则在现有的计算条件下无法找到 $m'$ ，使得 $h = H(m')$ 。
  - 即无法找到另外一个不同的报文，使得该报文的哈希值与已知报文的哈希值相同。



## 报文摘要与密码哈希函数(续2)

- 无冲突性：如果 $m_1 \neq m_2$ ，则 $H(m_1) \neq H(m_2)$ 。
  - 即不同的报文通过哈希运算，必须对应不同的哈希值。
- 满足以上3个特性的哈希函数就称为“密码哈希函数”。
- 互联网中常用的密码哈希函数是开源的报文摘要算法MD5，而目前公认较为安全的密码哈希函数是安全哈希算法SHA-1及其升级版本。



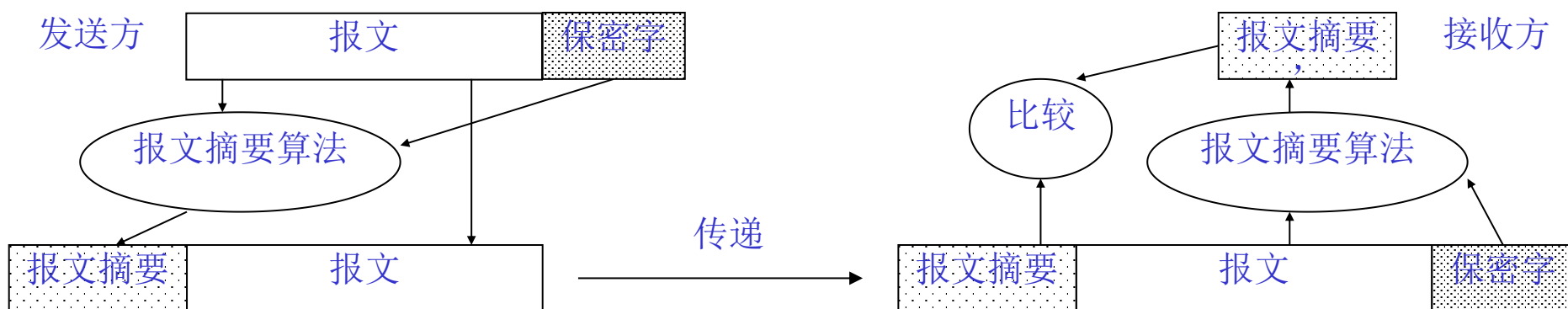
# 报文验证码

- 对报文摘要算法产生的报文摘要进行加密，或者报文摘要包含了双方约定的保密数据，可以得到用于报文真实性验证的代码，我们称为报文验证码(英文缩写为MAC)。
- 这种报文验证码可以验证该报文是真实的发送方发出的报文，因为真实的发送方才持有加密的密钥(发送方真实性验证)；
- 这种报文验证码也可以验证报文在传递过程中没有被修改(报文传递的完整性验证)。

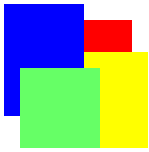


# 非加密报文验证方法

- 我们也可以不使用加密算法，利用报文摘要算法也可以生成报文验证码。即我们不使用解密算法，也可以进行报文验证。具体如下图所示。
- 注：需要发送方和接收方事先约定“保密字”

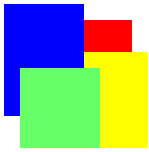


无加密报文验证方法



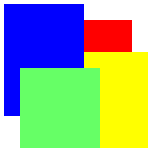
# 报文摘要算法MD5

- MD5是麻省理工学院（MIT）Ronald L. Rivest教授提出的一种报文摘要算法。该算法可以用于任何长度的报文M，生成128比特长度的、可以唯一标识报文M的“指纹”，即报文摘要。
- 报文摘要算法可以应用于报文真实性验证，也可以应用于报文的数字签名。
- 实际上数字签名也是一种报文真实性验证，数字签名是可以提供给第三方进行验证的报文验证方式。



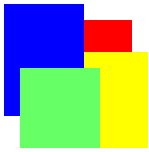
# MD5算法基本处理步骤

- 假定输入的报文长度是 $l$ 比特，MD5算法设置了4个字寄存器，最终存放在该4个字寄存器中的128比特的数就是报文摘要。
- MD5算法需要经过以下处理步骤：
  - 报文填充，
  - 填写报文长度，
  - 初始化寄存器，
  - 每次对每个512比特的报文块循环执行MD5算法，
  - 处理完全部报文后输出报文摘要。



# 安全哈希算法SHA-1

- 安全哈希算法(英文缩写为SHA-1)是一种类似于MD5的报文摘要算法。
- SHA-1算法首先扩展了报文摘要的长度，输入任何一个长度小于 $2^{64}$ 比特的报文，该算法可以输出长度为160比特的报文摘要。
- SHA-1算法比MD5具有更长的报文摘要，使得该算法比MD5算法更加安全。



# SHA-1算法的执行过程

- SHA-1算法执行的主要步骤与MD5算法的执行步骤基本相同，也包括以下步骤：
  - 填充报文，
  - 填写报文长度，
  - 初始化寄存器，
  - 循环计算报文块，以及
  - 输出报文摘要。



# 哈希报文验证码算法HMAC

- 哈希报文验证码(HMAC)提供了一种利用密钥构造报文验证码的标准方法:
- HMAC利用已有的密码哈希函数, 例如MD5和SHA-1, 在原来报文之前附加密钥(保密字), 构成“扩展报文”, 再通过密码哈希函数生成这种“扩展报文”的摘要, 构造一种可以用于报文真实性验证的MAC。
- HMAC使用的密钥不是数据加密中用于加密和解密的密钥, 这里“密钥”可以称为保密字。



# HMAC算法

- 假设待生成HMAC报文验证码的报文为M，使用的密码意义上的哈希函数为H，并且假定H是每次处理的数据块长度为B个字节(对于MD5和SHA-1，B = 64)，生成的哈希值长度为L个字节(对于MD5，L = 16，对于SHA-1，L = 20)。
- 假定用于报文验证的密钥长度K不大于B个字节，也不小于L个字节。即

$$L \leq K \leq B$$



## HMAC算法(续)

- HMAC定义2个固定的比特串：内填充值IPAD和外填充值OPAD如下：
- IPAD: 0011 0110 (十六进制: 0x36)重复B次。
- OPAD: 0101 1100 (十六进制: 0x5C)重复B次。
- HMAC算法公式表示如下：

$$\text{HMAC} = \text{H}(\text{K} \oplus \text{OPAD}) \parallel \text{H}(\text{K} \oplus \text{IPAD} \parallel \text{M})$$

- 这里“ $\oplus$ ”表示“异或”操作，“ $\parallel$ ”表示比特串合并操作，例如“0011  $\parallel$  0110”等于“00110110”，“0011  $\oplus$  0110”等于“0101”。



# HMAC的报文验证过程

- 报文接收方B采用HMAC验证从发送方A发送来的报文真实性的过程如下：
  - (1) 接收方收到报文后，按照报文传递协议的约定分离出报文部分M和报文验证码 $MAC_S$ 部分。
  - (2) 利用B方与A方约定的密钥K(保密字)和报文摘要算法(例如MD5或者SHA-1)，采用HMAC重新计算接收报文的报文验证码 $MAC_R$ 。
  - (3) 如果 $MAC_S = MAC_R$ ，则B方可以确信接收的报文确实是从A方发送来的，而且报文内容中途没有被篡改。



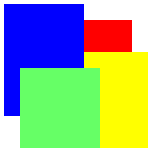
# 数字签名

- 数字签名是采用公钥加密算法中签名方的私钥，对电子文件的报文摘要加密生成的密文。  
数字签名技术 = 公钥加密技术 + 报文摘要技术
- 数字签名是公钥加密算法在报文验证技术中的具体应用。公钥加密算法最为成功的应用就是数字签名。
- 问题：能否采用传统加密算法进行数字签名？
- 采用传统加密算法的报文验证技术不能进行数字签名，这是因为传统加密算法的密钥是报文发送方A和接收方B共有的。



# 数字签名(续)

- 公钥数据加密有2种特性，使得它可以应用于这类彼此不信任的报文验证环境：
  - 其一：公钥数据加密将密钥分解成公钥和私钥两个部分，只有密钥所有者才持有私钥，其他人只可以获得公钥。
  - 其二：只要用公钥和私钥中任意一种密钥加密，就可以用另外一种密钥解密。
- 与手工签名相比，数字签名的优点在于签名与文件的内容相关。
- 数字签名的缺点在于：一旦私钥丢失或者泄漏，可能会被他人冒用而造成很大的损失。



# 身份真实性验证协议

- 身份真实性验证协议基本概念
- Needham-Schroeder身份真实性验证协议
- Needham-Schroeder协议的改进



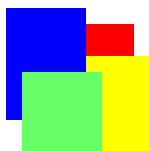
# 真实性验证协议的定义和作用

- 真实性验证协议是一种通过报文交互，验证交互的某一方或者交互双方身份真实性的协议。
  - 报文交互是网络参与方之间唯一的交互方式
- 只能验证报文交互某一方身份真实性的协议称为单向真实性验证协议，能够验证报文交互双方身份真实性的协议称为双向真实性验证协议。
  - 真实性验证协议的作用：（1）真实性验证协议可以验证身份真实性；（2）可以分发通过真实性验证的交互一方或者双方使用的密钥，或者协商密钥；（3）可以验证接收方收到的报文是否是正常传递的、而不是被截获后重发的报文，防范网络攻击者对真实性验证协议本身的攻击。



# 真实性验证协议基本方法和分类

- 真实性验证协议最早是由Roger M. Needham和Michael D. Schroeder提出的，其基本思想是利用加密方法在公共网络应用场景中进行身份真实性验证。
- 方法：目前真实性验证协议沿用了这种交互被加密报文的思路，实现网络环境下的真实性验证。
- 分类：可以根据采用的加密方法不同，真实性验证协议分成基于传统数据加密的真实性验证协议和基于公钥数据加密的真实性验证协议。



# 真实性验证协议的表示方式\*

- 国际网络安全中通常设置两个交互方为Alice（爱丽丝，简称为A）和Bob（鲍伯，简称为B），称为A方和B方。实际上A和B都是某个用户或应用的客户端。
  - 注：相当于中文俗称的“甲方”和“乙方”
- A方往B方发送一个报文，其中包括A的标识，以及采用密钥K加密的A的标识和一次性数N，具体表示如下：
$$M1: A \rightarrow B: A, K\{A, N\}$$
- 这里“ $K\{A, N\}$ ”表示采用密钥K对A和N加密后的密文。



# 传统数据加密真实性验证协议原理

- 基于传统加密算法的真实性验证协议的原理是：如果一个当事方能够正确地利用某个密钥加密数据，并且验证方相信只有身份标识对应的当事方才知道这个密钥时，则验证方就可以确信真实性验证协议的交互方是具有该身份标识的当事方。
- 例如假定A试图向B验证身份，A的标识就表示为“A”，而且 $K_{A,B}$ 是B和A公共拥有的密钥，一个简单的真实性验证协议如下：

M1:  $A \rightarrow B: A, K_{A,B}\{A\}$



# 传统数据加密真实性验证协议-1

- 在以上真实性验证协议中存在一个致命的弱点，就是无法防范网络攻击者的重播攻击。即：如果网络攻击者C可以截获报文M1，等到A方离开网络后，C再重发报文M1。这样，B就会误认发送报文M1的C就是A，C就可以假冒A与B进行交互。

报文M1的重播攻击:  $M1: C \rightarrow B: A, K_{A,B}\{A\}$

- 为了防范重播攻击，需要对以上真实性验证协议进行改进，引入可以表示交互的报文已经使用过的标志，这种标志在真实性验证协议中称为一次性数，表示为N。



# 基于一次性数真实性验证协议\*

- 一次性数是一种在报文中仅仅使用一次的随机数，为了方便对一次性数的验证，通常由验证方产生一次性数（例如：登录时输入的验证码就是一次性数，由服务器产生，用于防范假冒的登录方）。引入一次性数的真实性验证协议如下：

M1:  $A \rightarrow B: A$

M2:  $B \rightarrow A: N$

M3:  $A \rightarrow B: K_{A,B}\{N\}$

- 以上基于传统加密法的、采用一次性数的真实性验证协议是一个经典的真实性验证协议。



## 三方参与的真实验证协议\*

- 由于以上真实验证协议没有密钥 $K_{A,B}$ 协商过程，所以，该协议无法一个在大规模网络环境下使用。
- 验证方不可能与所有可能的当事方都事先协商好密钥。为了解决在大规模网络中真实验证协议的可缩放性问题，就需要在真实验证协议中引入第三方：真实性验证服务器(AS注册服务器)。
  - 这就是为何安全访问网站的第一步是“注册”
- 真实性验证服务器中存放了它与所有注册用户(当事方)的对称密钥，例如 $K_{AS,A}$ 表示AS与A之间的密钥，而 $K_{AS,B}$ 表示AS与B之间的密钥。



## 三方参与的真实验证协议(续1)\*

假设A和B已经在AS服务器上注册，则引入真实验证服务器和一次性数的真实验证协议如下：

(1) A向B发送一个包含A的标识的报文：

M1:  $A \rightarrow B: A$

(2) B向A返回包含一次性数的报文：

M2:  $B \rightarrow A: N$

(3) A向B发送包含 $K_{AS,A}$ 加密的一次性数的报文：

M3:  $A \rightarrow B: K_{AS,A}\{N\}$



## 三方参与的真实验证协议(续2)\*

(4) B向AS发送包含 $K_{AS,A}$ 加密的一次性数的报文:

M4:  $B \rightarrow AS: A, K_{AS,A}\{N\}$

(5) AS解密采用 $K_{AS,A}$ 加密的一次性数, 并返回包含采用 $K_{AS,B}$ 加密的一次性数报文:

M5:  $AS \rightarrow B: K_{AS,B}\{N\}$

- 部署以上真实验证协议的前提条件:
  - (i) 当事方与验证方有共同信任的真实验证服务器AS。
  - (ii) 当事方、验证方都必须在真实验证服务器注册。即已经与AS协商好密钥 $K_{AS,A}$ 和 $K_{AS,B}$ 。



# 公钥数据加密真实性验证协议

- 基于公钥数据加密的真实性验证协议的原理：如果一个当事方能够正确地利用某个身份标识对应的私钥进行数字签名，则验证方就可以确信真实性验证协议交互的数字签名方是具有该身份标识的当事方。
- 这里的“数字签名”表示采用公钥数据加密中的私钥对某个特征数(例如一次性数)进行的加密运算。
- 具有网上交易权限的网上银行用户的身份真实性验证可以采用这类协议，因为这类用户具有私钥。但网银用户需要先注册！



## 公钥数据加密真实性验证协议 (续1)

- 假定当事方A需要向验证方B验证自己的身份，A的私钥为 $VK_A$ ，并且验证方B已经获得了与 $VK_A$ 对应的公钥 $PK_A$ ，则基于公钥数据加密的、采用一次性数的真实性验证协议如下：

M1: A  $\rightarrow$  B: A

M2: B  $\rightarrow$  A:  $N$

M3: A  $\rightarrow$  B:  $VK_A\{N\}$

- 在以上两方真实性验证协议中，假定B可以从权威的证书授权中心(CA)获得A的公钥 $PK_A$ 。



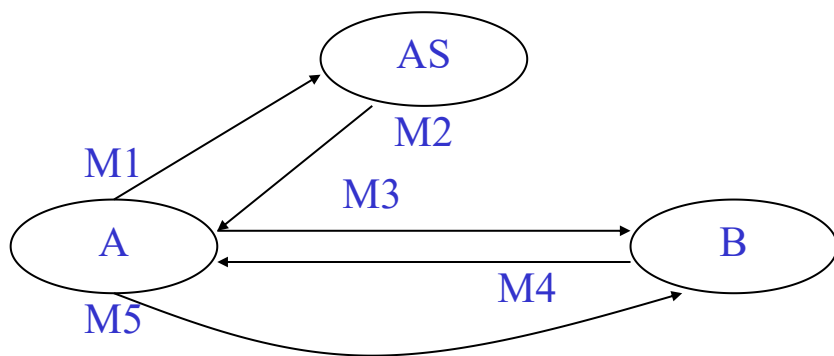
# Needham-Schroeder真实性验证协议

- Needham-Schroeder真实性验证协议假定是在一个不安全的网络环境下，并且假定真实性验证的A和B之间存在一个双方都信任的真实性验证服务器AS，并且A和B分别与AS已经约定了密钥 $K_{AS,A}$ 和 $K_{AS,B}$ 。
- A和B试图通过真实性验证协议，相互确认对方身份，并且建立双方共有的密钥 $K_{A,B}$ ，使得双方可以利用该密钥采用传统数据加密算法加密传递数据。
- 这里密钥 $K_{A,B}$ 通常也称为传递数据的会话密钥。



# N-S真实性验证协议交互过程\*

- 与设计传统的计算机网络协议的规则一样，Needham-Schroeder真实性验证协议也假定不依赖任何精确的全球时钟系统，而是通过报文的异步交互实现双方的真实性验证。该真实性验证协议的交互涉及5个报文，具体交互过程如下所示



AS: 真实性验证服务器

M1:  $A, B, N_A$

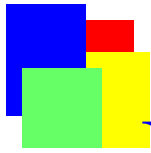
M2:  $K_{AS,A} \{N_A, B, K_{A,B}, K_{AS,B} \{K_{A,B}, A\}\}$

M3:  $K_{AS,B} \{K_{A,B}, A\}$

M4:  $K_{A,B} \{B, N_B\}$

M5:  $K_{A,B} \{A, N_B - 1\}$

Needham-Schroeder真实性验证协议



# Needham-Schroeder协议交互过程

- Needham-Schroeder协议可以表示为如下形式：

M1:  $A \rightarrow AS: A, B, N_A$

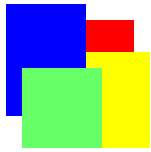
M2:  $AS \rightarrow A: K_{AS,A} \{N_A, B, K_{A,B}, K_{AS,B} \{K_{A,B}, A\}\}$

M3:  $A \rightarrow B: K_{AS,B} \{K_{A,B}, A\}$

M4:  $B \rightarrow A: K_{A,B} \{N_B\}$

M5:  $A \rightarrow B: K_{A,B} \{N_B - 1\}$

- 这里可能出现的问题是在报文M3，这里没有任何有关M3报文一次性使用的标记。



# Needham-Schroeder协议的改进

- 对Needham-Schroeder协议存在的问题有两种改进方案：
  - (1) 基于时间戳的改进方案
  - (2) 基于一次性数的改进方案



## 基于时间戳的改进方案\*

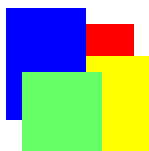
- Denning和Sacco提出了在Needham-Schroeder协议中增加时间戳的方案，主要修改了以上协议报文M1、M2和M3

M1':  $A \rightarrow AS: A, B$

M2':  $AS \rightarrow A: K_{AS,A} \{T, B, K_{A,B}, K_{AS,B} \{K_{A,B}, A, T\}\}$

M3':  $A \rightarrow B: K_{AS,B} \{K_{A,B}, A, T\}$

- $T$ 是时间戳，标记AS在确定密钥 $K_{A,B}$ 时的本地时间。利用这个时间戳，A和B都可以验证AS返回A的报文以及A发送给B的报文是否是重播攻击报文。

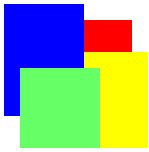


## 基于时间戳的改进方案(续)

- 假定Clock表示A或者B接收到报文M2'或者M3'的时间，A或者B可以利用以下公式验证M2'或者M3'是否是重播攻击报文：

$$| \text{Clock} - T | < \Delta t_1 + \Delta t_2$$

- 这里 $\Delta t_1$ 表示服务器时钟与A或者B时钟的最大预期误差， $\Delta t_2$ 表示报文在网络中传播的最大预期延迟。对于B， $\Delta t_2$ 还需要包括在A中处理报文M2'的延迟。
- 这里的时钟可以采用计算机系统时钟， $\Delta t_1$ 可以设置为1分钟。
  - 注：交互双方的时间同步在安全管理中十分关键！安全性要求高的网络，可以采用NTP(网络时间协议)实现时间同步。



# 基于一次性数的改进方案\*

- 真实性验证协议的最初提出者Needham和Schroeder不赞同基于时间戳的改进方案，提出了基于一次性数的改进方案。
- 他们认为，这种改进破坏了互联网中绝大部分协议遵循的一个基本原则：不依赖于全球统一时钟进行交互。
- 基于一次性数的改进方案新增加了A和B之间的2次交互，使得A首先从B中获得一次性数。



## 基于一次性数的改进方案(续)

- 这种基于一次性数的改进(标红部分)协议如下:

M1: A  $\rightarrow$  B: A

M2: B  $\rightarrow$  A:  $K_{AS,B}\{A, N'_B\}$

M3: A  $\rightarrow$  AS: A, B,  $N_A$ ,  $K_{AS,B}\{A, N'_B\}$

M4: AS  $\rightarrow$  A:  $K_{AS,A}\{N_A, B, K_{A,B}, K_{AS,B}\{K_{A,B}, A, N'_B\}\}$

M5: A  $\rightarrow$  B:  $K_{AS,B}\{K_{A,B}, A, N'_B\}$

M6: B  $\rightarrow$  A:  $K_{A,B}\{B, N_B\}$

M7: A  $\rightarrow$  B:  $K_{A,B}\{A, N_B - 1\}$

- 理论上, 基于一次性数的改进协议是比较完美的一种协议。只是这种协议增加了A和B的交互次数 b40



# 公钥基础设施PKI

- PKI的必要性
- PKI的结构
- 证书与X.509建议
- PKI的实现模型
- PKI的设计建议



# PKI的必要性

- 如下是一个典型的电子商务交互过程:

M1:  $A \rightarrow B: PK_B\{\text{订单}, VK_A\{\text{签名}\}\}$

M2:  $B \rightarrow A: PK_A\{\text{回执}, VK_B\{\text{签名}\}\}$

M3:  $A \rightarrow B: PK_B\{\text{确认}, VK_A\{\text{签名}\}\}$

- 这是否一定能够保证以上交易的真实可信?

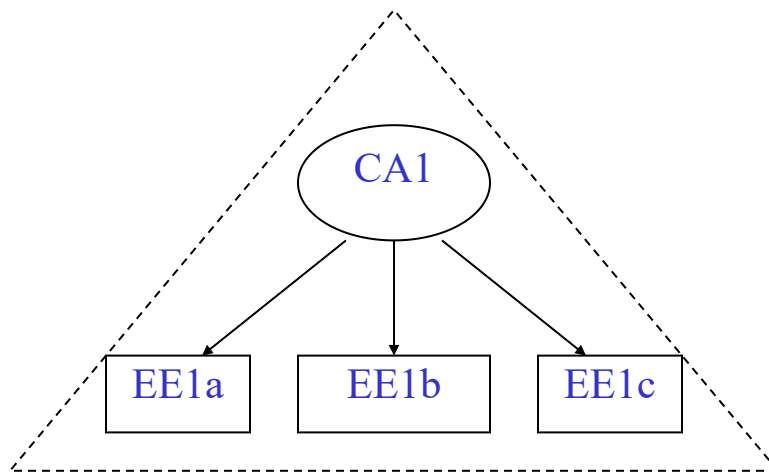
**问题:** 虽然通过公钥 $PK_A$ 和 $PK_B$  才能保证交易的真实可信, 但公钥的真实性是如何保证的?

**答案:** 必须设计一套公钥权威发布和更新的系统: 公钥基础设施(英文缩写PKI)



# PKI的结构\*

- PKI通常采用信任金字塔(POT)结构。最简单的POT只有两层结构，CA(认证权威中心, Certification Authority)处于POT的塔尖，而由该CA签署发布的EE(端实体)证书处于POT的塔底。
- 这种基本的POT结构就构成了PKI中一个基本的信任域。

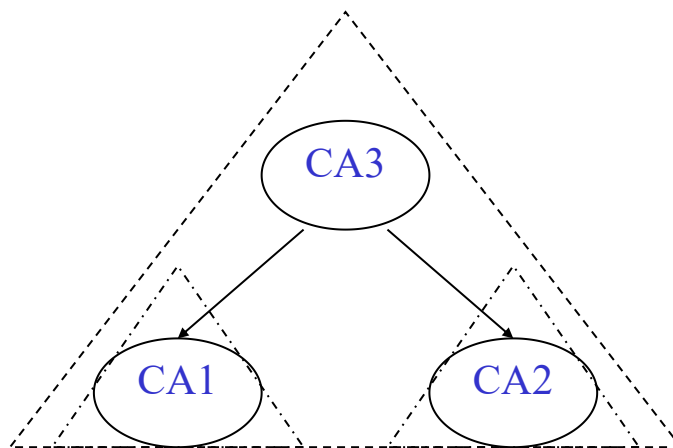


PKI信任金字塔 (POT) 结构



## PKI的结构(续1)\*

- 在PKI具体实现结构中，通常需要涉及到多个信任域之间的证书获取和验证，单个信任域的PKI结构就无法满足要求。这时，可以通过多层POT结构，实现跨信任域的证书获取和验证。

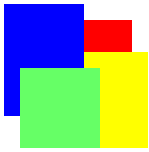


PKI多层信任金字塔(POT)结构



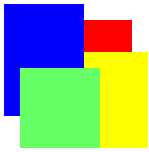
# 证书的定义与逻辑结构\*

- 证书是绑定公钥与某个实体标识的一种数据结构，并且由使用该证书的实体信任的认证中心CA签名发布的。常用的证书一般包括证书标识、证书发布方信息、证书持有方信息、证书使用信息，具体包括以下内容：
  - (1) 证书编号，这是CA发布的唯一编号。
  - (2) 证书发布方名字。
  - (3) 证书持有方名字。
  - (4) 证书持有方的公钥。
  - (5) 计算该证书数字签名的算法。
  - (6) 证书有效期。
  - (7) 证书发布方签名，以及其他证书选项。



# 证书与X.509建议\*

- 以上这种证书的结构是国际电信联盟电信标准化部门(ITU-T)发布的X.509建议中定义的证书结构，这是目前国际上标准的证书格式。
- 更新版的X.509建议是2000年3月由ITU-T正式批准的文本。
- X.509定义了一个公钥证书的框架模型，它包括用于描述证书的数据对象规范，以及对已经发行的证书发布不再信任的作废公告规范。



## X.509建议证书的特征\*

- X.509建议定义的公钥证书具有以下特征：
  - 其一，任何属于某个认证权威中心公钥的用户，都可以从该认证权威中心发行的证书中获取公钥；
  - 其二，除了发行证书的认证权威中心之外，任何个人或机构都无法修改证书而不被察觉。



## X.509建议证书的格式\*

- 按照X.509建议的定义，认证权威中心CA可以签名一组信息组成的某个用户的证书。该组信息包括用户名A和公钥 $PK_A$ ，以及包括该用户的唯一标识符UA等。

$$CA[A] = VK_{CA} \{V, SN, AI, CA, UCA, A, UA, PK_A, TIME_A\}$$

- V: 版本号，SN: 证书序号，AI: 数字签名算法，
- CA: 发布方名称，UCA: 发布方唯一标识符，
- A: 用户名称，UA: 用户唯一标识符，
- $PK_A$ : 用户公钥信息， $TIME_A$ : 公钥有效期，
- $VK_{CA}$ : 发布方私钥，用于发布方的数字签名。



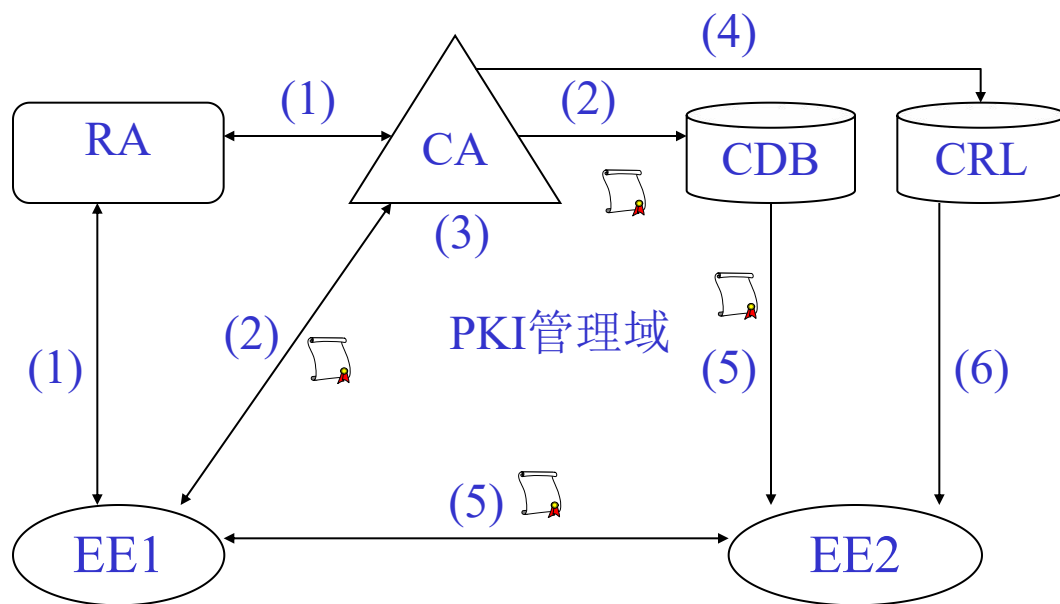
# PKI的实现模型\*

- 原理上，PKI的实现模型一般包括以下几个部分：
  - 可信的发布公钥的**认证权威中心（CA）**，
  - 证书持有者注册的**注册权威中心（RA）**，
  - 存放有效证书的**证书数据库（CDB）**，
  - 存放作废证书的**证书作废表（CRL）**，以及
  - 使用PKI服务的**端实体（EE）**。
- **认证权威中心(CA)**通常将注册用户的**身份真实性验证、密钥对生成**等操作交给**注册中心**处理，而证书的**签署、发布、作废**等关键操作由**CA**处理。
- **CA**直接**管理和操纵证书数据库和证书作废表**。



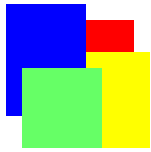
## PKI的实现模型(续)\*

证书的发布和使用包括的处理：(1)注册与密钥对的生成阶段。(2)证书的产生和分发。(3)证书过期与更新。(4)证书作废。(5)证书获取(发布)。(6)证书验证(查找黑名单)。



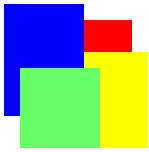
RA: 注册权威中心  
CA: 认证权威中心  
CDB: 证书数据库  
CRL: 证书作废表  
EE: 端实体  
📄: 证书

一种PKI实现模型



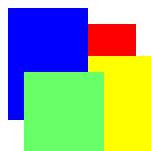
# 访问控制技术与应用

- 访问控制基本概念
- 自主访问控制策略与访问控制列表
- 强制访问控制策略与Bell-LaPadula模型
- 交易访问控制策略与Clark-Wilson模型
- 基于角色的访问控制
- 网络防火墙基本概念\*
- 网络层防火墙\*
- 应用层防火墙



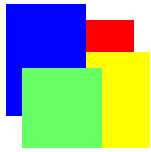
# 访问控制基本概念

- 访问控制策略，
- 访问控制机制，
- 访问控制模型，
- 访问控制的主体和客体，
- 托管监控器模型。



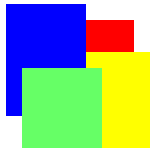
# 访问控制策略\*

- 访问控制策略表示访问控制的总体要求(包括约束条件)。
- 访问控制策略描述了进行访问控制的规则，明确了哪些用户、在何种环境下、可以访问哪些信息。
  - 例如用户A可以访问文件服务器B中的目录C下的所有文件。
- 典型的访问控制策略包括：自主访问控制(DAC)策略，强制访问控制(MAC)策略，等等。



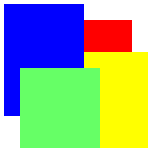
# 访问控制机制\*

- 访问控制机制是对访问控制策略的具体实现，它可以表示为一组硬件或者软件的访问控制实现方法。
- 典型的访问控制机制是访问控制列表和访问控制矩阵。
  - 例如：采用访问控制列表(ACL)可以实现对不同用户设置不同的访问文件系统的控制策略。



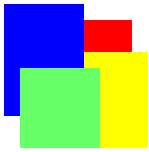
# 访问控制模型\*

- 为了能够较为完整地研究系统的访问控制能力，需要构建访问控制模型。
- 访问控制模型描述访问控制系统的安全控制规则，以及对访问控制策略的支撑规则。
  - 注意区别访问控制规则
- 典型的访问控制模型包括： Bell-LaPadula模型(安全控制规范)、 Clark-Wilson模型(安全控制规范和支撑规范)等。



# 主体\*

- 在信息安全中，代表用户访问网络或使用网络应用的某个实体是访问控制的“主体 (Subject)”。
- 在网络安全中，代表用户访问网络或使用网络应用的任何实体都是访问控制的主体，这是访问控制中被控制方。
  - 例如电子邮件的客户端软件、万维网 (WWW) 客户端软件、文件传送系统的客户端软件
- 主体是访问控制的请求方。



# 客体\*

- 在信息安全中，任何被访问网络的实体或被使用网络应用的实体都是访问控制的“客体(object)”。
- 在网络安全中，任何提供网络服务或网络应用的实体都是客体。
  - 例如，万维网（WWW）服务器、文件服务器等。
- 客体是访问控制的被保护方。



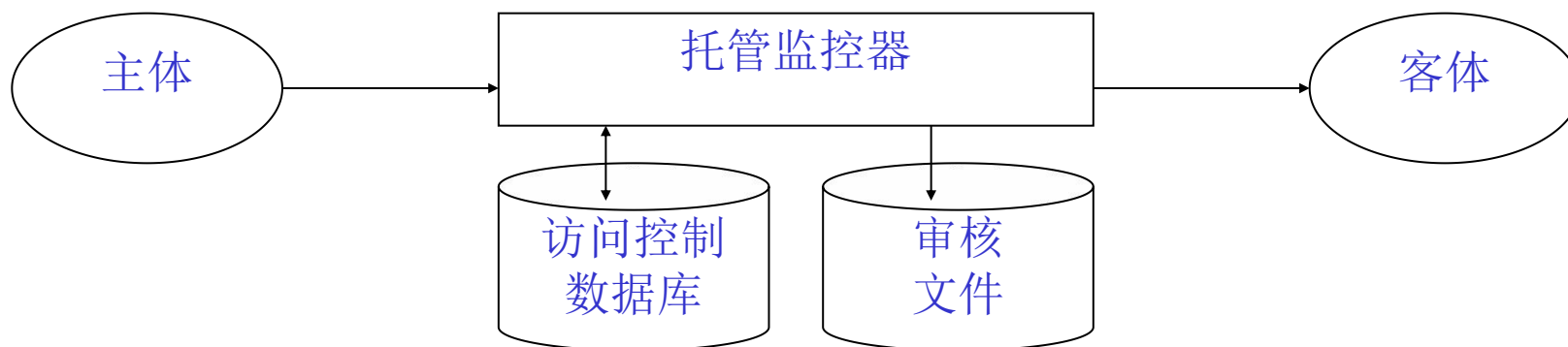
# 托管监控器模型\*

- 主体和客体之间需用一个访问控制实体。
- J. P. Anderson于1972年提出的“托管监控器”通常作为访问控制的框架模型。
- 它可以既作为表示高可信的访问控制实体所必备功能单元的一种抽象框架模型，又可以作为设计、实现和分析安全信息系统的一个参考模型。
- 信息系统中任何一个主体需要通过托管监控器的访问权限审核之后，才能访问相应的客体。
- 实现“托管监控器”的功能模块称为“安全内核”



# 托管监控器模型的特性

- 为了保证托管监控器能够按照访问控制规则，实现访问控制策略，托管监控器必须具备3个特性：
  - 完备性(任何访问操作无法绕过托管监控器)，
  - 孤立性(独立安全机制、不依赖其他系统、不受其他系统干扰)，
  - 可验证性(控制机制及其实现必须通过安全验证)。

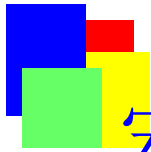


托管监控器访问控制架构



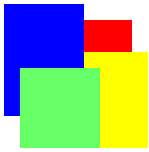
# 托管监控器模型的应用

- 托管监控器作为一种访问控制系统的抽象框架结构，在30多年来一直指导信息通信系统的安全设计、实现和评价。
- 根据托管监控器模型可以得出设计访问控制系统的3条原则：
  - (1) 灵活性，能够实现任何访问控制策略
  - (2) 可管理性，易于配置和管理
  - (3) 可缩放性，适合任何规模的应用环境



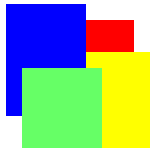
# 军用安全策略与Bell-LaPadula模型

- 军用安全策略是指美国国防部在1983年发布的“可信计算机系统评价准则 (TCSEC)”中提出的自主访问控制 (DAC) 策略和强制访问控制 (MAC) 策略。
- 军用安全策略主要用于数据保密。
- 为了实现MAC策略，必须采用Bell-LaPadula模型。



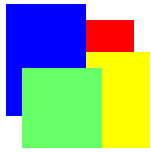
# 自主访问控制策略\*

- 自主访问控制（DAC）策略是基于用户的标识，或者基于用户所属组的标识，限制对客体访问的一种方式。
- 在DAC策略中，用户（主体）自己拥有的资源（客体）的访问权限可以自主地传递给其他用户（主体）。
- DAC是所有访问控制系统中必须支持的一种访问控制策略。



# 自主访问控制策略的实现\*

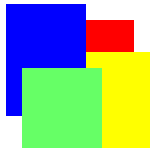
- 通常采用基于客体的访问控制列表（ACL）机制实现DAC策略。
- 为了提供对DAC的支持，需要明确定义资源（客体）的拥有者。
- 只有资源拥有者可以修改该资源的ACL，并可以向其他用户传递访问控制权限。这样，要求每个客体维护一张访问控制列表（ACL）。



# 自主访问控制策略的实现举例1

**例4.1：** 如果假定用户A1可以“读/写”访问目录D1和D2，A2用户可以“读/写”访问目录D2，则目录D1的访问控制列表（ACL）表示如下：

主体	客体	控制	操作
A1	D1	允许	读/写



## 自主访问控制策略的实现举例2

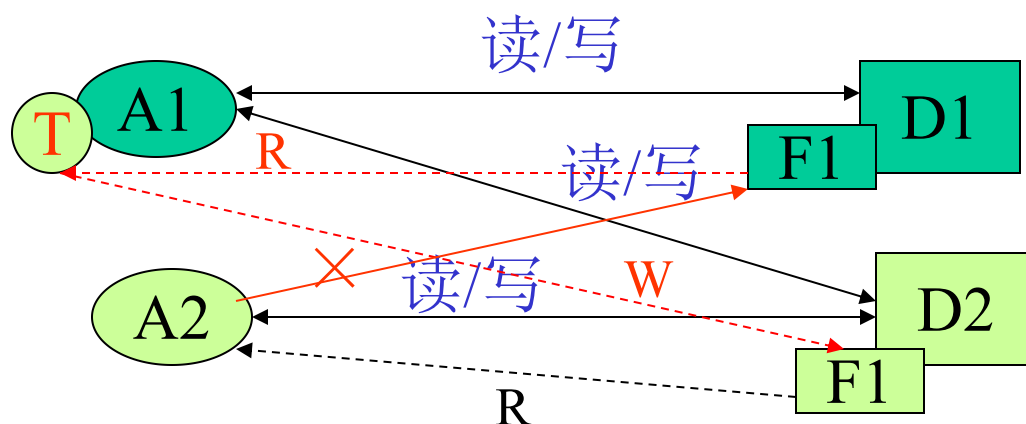
- 目录D2的访问控制列表（ACL）表示如下  
（注：没有明确允许的访问控制默认为禁止）：

主体	客体	控制	操作
A1	D2	允许	读/写
A2	D2	允许	读/写



# 自主访问控制策略的潜在威胁\*

- DAC访问控制策略可能遭遇木马的攻击。
- 在前面的举例中，如果A2将木马程序T驻留在A1的运行主机中，则T可以首先读取目录D1的文件F1，然后写入目录D2中，这样A1就可以读取D2中的F1。



T-R-D1.F1 →  
T-W-D2.F1 →  
A2-R-D2.F1



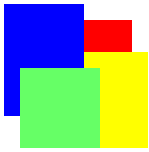
## 自主访问控制策略的弱点\*

- DAC控制策略存在本质上的安全弱点：
  - (1) 无法区别资源的产生者与资源真正的拥有者，DAC无法防范访问控制权限的传递。
  - (2) DAC也无法防范“特洛伊木马”的攻击。
- 对于安全控制要求较高的系统，必须采用其他更强的安全控制策略，例如强制访问控制策略(MAC)。



## 强制访问控制策略\*

- 在强制访问控制（MAC）中，为每个用户和可能被访问的资源都指派了安全等级。
- 安全等级包括层次化等级和非层次化等级。
- 层次化(等级化)部分包括：无限制(U)、秘密(C)、机密(S)、绝密(TS)。
- 非层次化(功能域)部分包括：国家安全部、核设施、军事设施、民用设施等。



## 强制访问控制策略(续)\*

- 提出MAC控制策略的一个目的是防范“特洛伊木马”的攻击。
- 访问控制列表无法实现MAC策略。为了实现MAC策略，必须采用Bell-LaPadula模型。



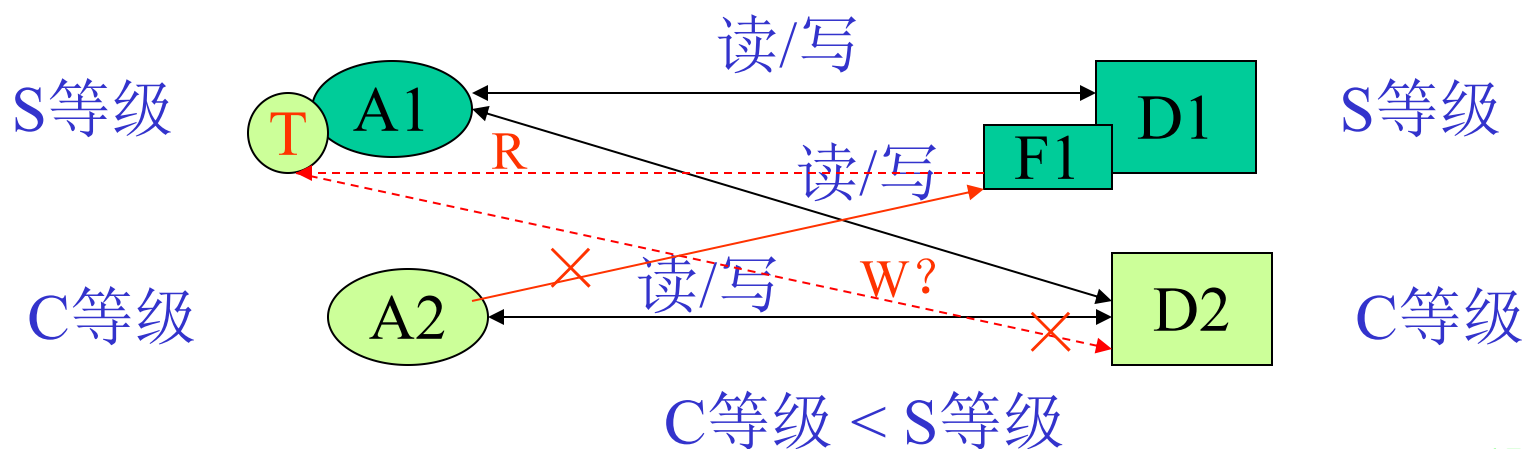
# Bell-LaPadula模型\*

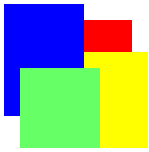
- Bell-LaPadula模型可以实现MAC策略的访问控制决策的两条规则如下：
  - 注意该模型的访问控制细分成为“读”访问控制和“写”访问控制，细分才能解决问题！
  - (1) 简单安全特征规则：任何一个主体只能“读”访问不大于其安全等级的客体
  - (2) 星状特征规则：任何一个主体只能“写”访问不小于其安全等级的客体。



## Bell-LaPadula模型优点\*

- MAC控制策略具有以下优点：
  - (1) MAC可以防范访问控制权限的传递。
  - (2) MAC可以防范“特洛伊木马”的攻击。





# 访问控制模型举例

假设某网络系统限定用户A1可以“读/写”访问目录D1和D2中的文件，用户A2可以“读/写”访问目录D2中的文件。

问题：

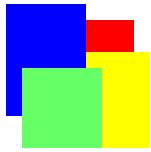
- (1) 采用DAC策略实现以上访问控制。
- (2) 采用MAC策略实现以上访问控制。
- (3) 如果A1拥有目录D1，A1是否可以将访问权限传递给A2？如果可以，如何传递？



## 访问控制模型举例(续1)

解答（1）：可以采用以下访问控制列表实现满足DAC策略的以上访问控制：

主体	客体	控制	操作
A1	D1	允许	读/写
A1	D2	允许	读/写
A2	D2	允许	读/写



## 访问控制模型举例(续2)

以上访问控制列表也可以表述为：

A1允许“读/写” D1； A1允许“读/写” D2；  
A2允许“读/写” D2。

**问题（2）** 采用MAC策略实现以上访问控制。

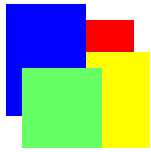
**解答（2）**：首先定义A1安全等级为S， A2安全等级为C， D1安全等级为S， D2安全等级为C， 并且 $S > C$ 。



## 访问控制模型举例(续2)

根据B-L访问控制模型可以得出实现满足MAC策略的以上访问控制的访问控制列表：

主体	客体	控制	操作
A1	D1	允许	读/写
A1	D2	允许	读
A2	D2	允许	读/写



## 访问控制模型举例(续3)

以上访问控制列表也可以表述为：

A1允许“读/写” D1； A1允许“读” D2；  
A2允许“读/写” D2。

**问题（3）** 如果A1拥有目录D1， A1是否可以将访问权限传递给A2？ 如果可以，如何传递？

**解答（3）：** 只有在DAC策略下， A1才能向A2传递对D1的访问控制权限。



## 访问控制模型举例(续4)

A1只需要在对D1的访问控制列表中增加以下控制规则：

主体	客体	控制	操作
A2	D1	允许	读/写



# 商用安全策略与Clark-Wilson模型

- D. Clark和D. Wilson首先指出了商用安全策略与军用安全策略的不同之处，指出人们在商业交易中更加关心的是交易的完整性，而不是交易的保密性。

- 注意：银行或网络支付的访问控制不再是数据保密！

传统商用领域控制欺诈和错误的两条基本原则：

- 原则1：采用严格的步骤、并可以事后审计的、完整的“正规交易”原则 - 程序规范
- 原则2：对于一个交易必须有多个雇员参与、可以相互监督、相互约束的“职责分离”原则。



# Clark-Wilson模型

- 为了实现商用安全策略，Clark和Wilson提出了一种访问控制模型，即Clark-Wilson模型。
- 该模型由一组规则构成，这组规则可以分成认证类规则（C(certification)类规则）和实施类规则（E(execution)类规则），主要用于实现“正规交易”策略和“职责分离”策略。
  - C类规则提供身份和数据真实性验证
  - E类规则具体实施身份和数据访问控制



# 引入RBAC的必要性\*

- 按照TCSEC规范设计的访问控制，无法适用于企业和政府文职机构的需求。
  - 企业和政府文职机构的职员仅仅使用机构的信息资源，并不拥有这些数据资源的所有权。另外，企业或政府机构都有副职临时代替正职行使权力的规定。。
- 按照DAC模型，应该将数据资源的所有权赋予职员，这是不合适的(不满足知识产权管理的需求)。
- MAC模型只强调对数据的保密，固定设置一个密级，这种安全控制要求无法满足对于信息处理的多样性和灵活性的需求。



# RBAC的基本原理\*

- 为了解决这些问题，在NIST工作的D. Ferraiolo和D. Kuhn在总结当时提出的各类面向应用的访问控制模型的基础上，于1992年提出了一种通用的基于角色的访问控制模型（英文缩写RBAC）。
- RBAC的基本原理：按照用户在某个机构中的“角色”，控制其对计算机系统中资源的访问。
- 作为一个通用的访问控制模型，RBAC需要完整地定义“用户”、“角色”和对“客体”的“交易”之间的关系，实现访问控制策略。
  - 问题：RBAC中的“用户”和“角色”哪个是主体？



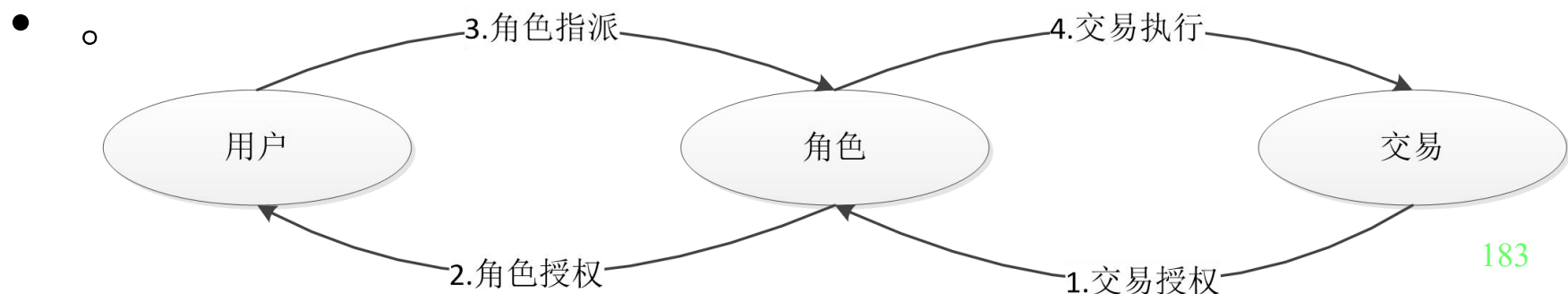
# RBAC原理分析\*

- 角色成为整个访问控制模型的核心。角色对应于企业和政府机构的“工作岗位”。
- 企业或政府机构的组织结构确定，工作岗位及其功能也确定。工作岗位上的工作人员可能常会流动。
- 完整、严格地设计好RBAC模型的角色和交易之间的授权关系，就可以完整、严密地进行动态访问控制。
- 工作人员的流动，或职责调整，只需进行工作人员的角色指派或角色授权的更改(人力资源部完成)，不需要涉及到对信息资源访问权限的重新评估。



# RBAC基本规则\*

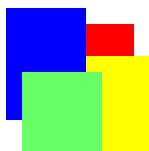
- 交易授权规则要求某个活跃角色只有被某个交易授权之后，该角色对应的用户才能执行该交易。
- 角色授权规则要求一个活跃的角色授权给用户之后，用户才能按照该活跃角色的权限执行相应的“交易”
- 角色指派规则要求所有用户只有被指派一个角色(访问控制模型中的“主体”)之后才能执行某个交易。
  - 这里“交易(事务处理)”表示对某类数据资源的操作。





# 网络防火墙定义

- 从理论上定义，防火墙是限制数据在网络之间自由传递的控制系统，它是网络系统中一种安全访问控制系统。
  - 网络层就是限定分组的传递
  - 应用层就是限定报文的传递
- 在实际应用中，防火墙是限制数据在企业内部网络、企业外部网络以及公共互联网之间自由传递的安全控制系统。



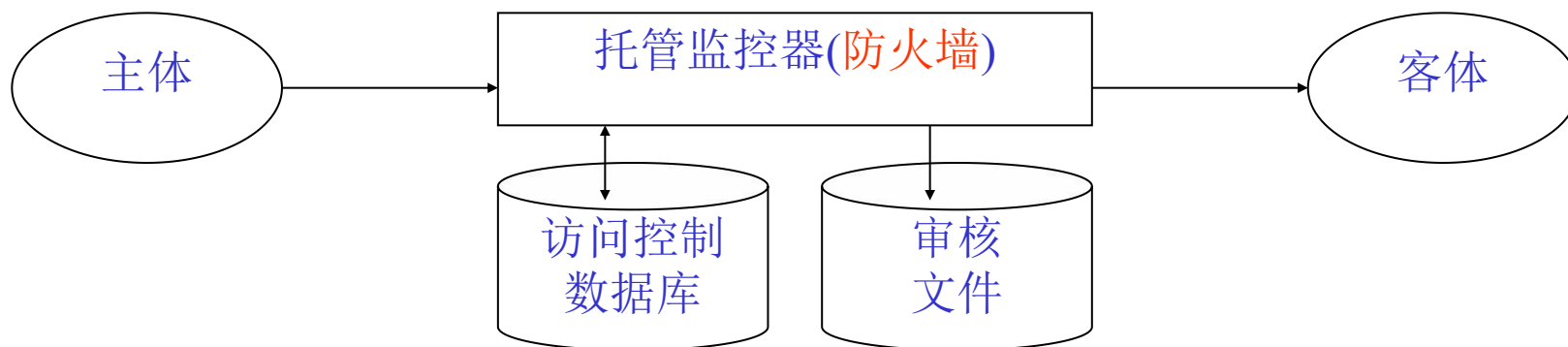
# 网络防火墙特征

遵循托管监控器原理，防火墙具有以下特征：

- (1) 所有从外部网到内部网，或者从内部网到外部网的分组流或报文流必须经过防火墙；
  - 完备性，不可翻墙！
- (2) 根据本地安全策略定义，只有被授权的分组流或报文流才能通过防火墙；
  - 孤立性，本地处理
- (3) 防火墙具有抵御网络攻击的能力。
  - 需要验证的能力，特别是机器验证能力！

# 托管监控器对防火墙的要求

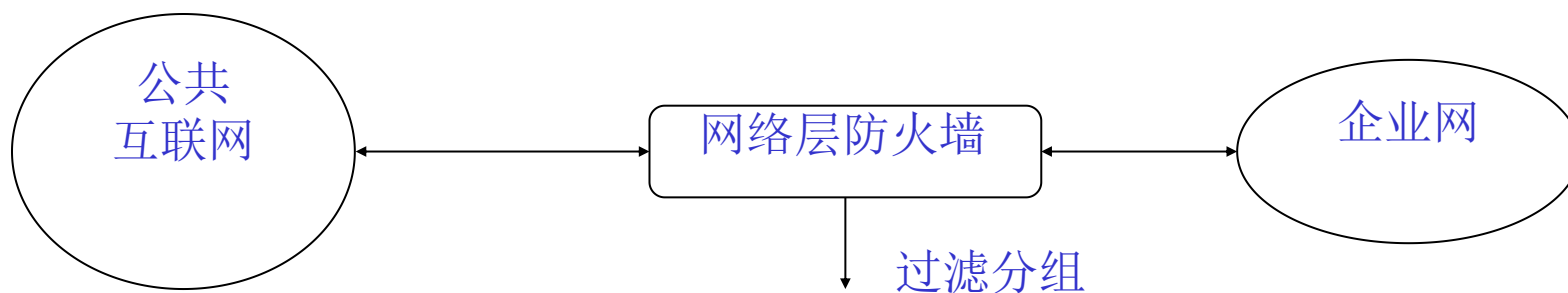
- 为了保证托管监控器能够按照访问控制数据库的规则，实现访问控制策略，基于托管监控器的防火墙必须满足3点要求：
  - 完备性(任何访问操作无法绕过) → 必须经过防火墙
  - 孤立性(独立安全机制、不受其他系统干扰) → 本地安全策略
  - 可验证性(控制机制及其实现必须通过安全验证) → 抵御攻击



托管监控器访问控制架构

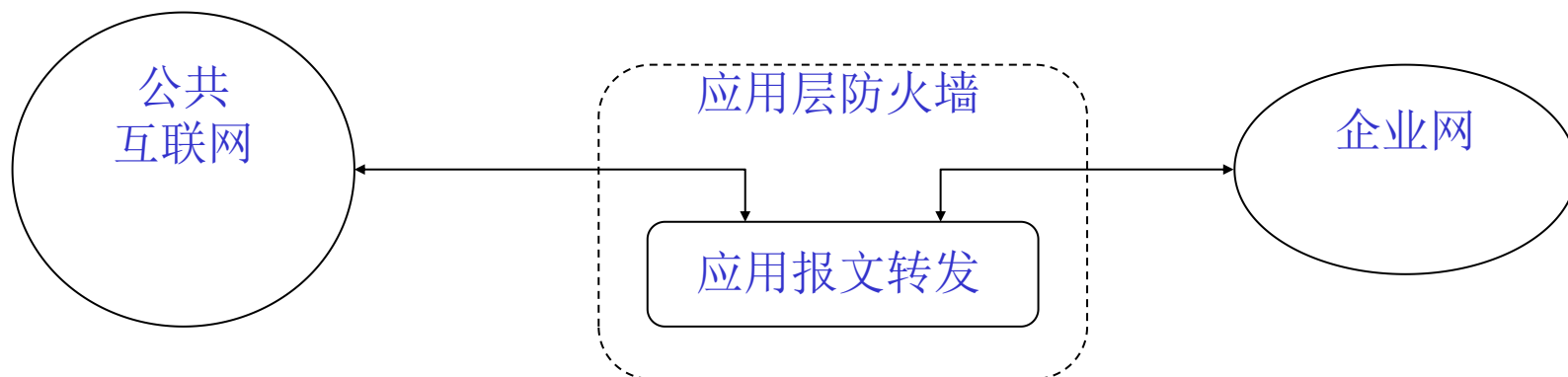
# 网络层防火墙

- 防火墙一般可以分成2种类型：网络层防火墙和应用层防火墙。
- 网络层防火墙也称为“分组过滤器”，它是通过网络路由交换设备对分组头的识别，过滤不符合安全策略的分组，实现对进入或者离开企业网的分组进行访问控制——路由交换设备的基本功能。



# 应用层防火墙

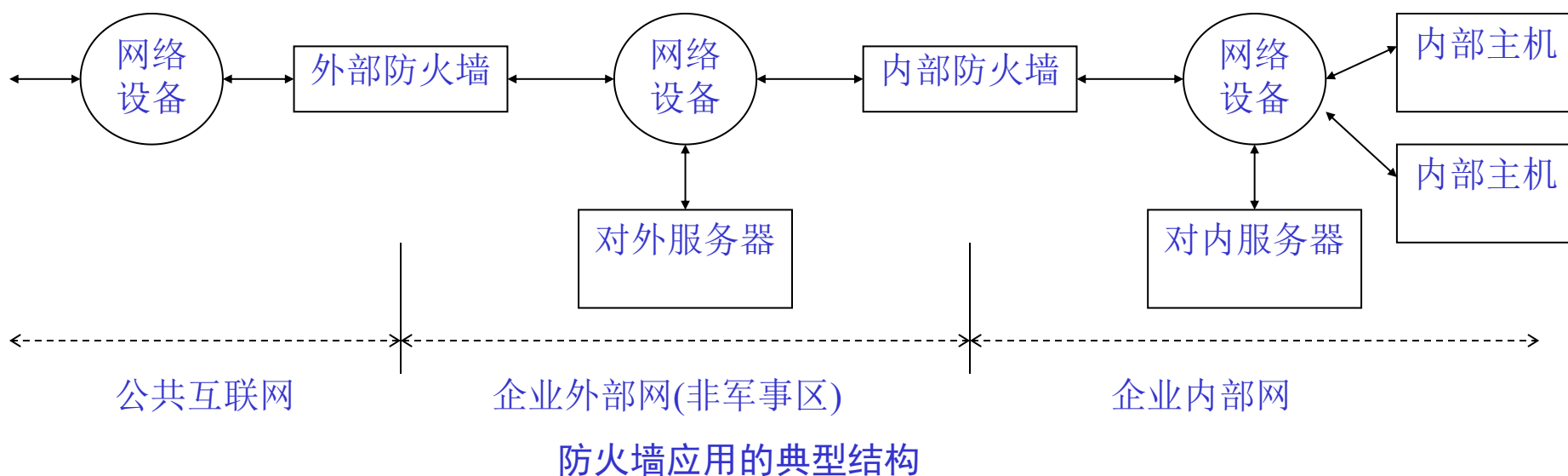
- 应用层防火墙也称为“报文转发器”或“应用网关”，它强制性在公共互联网与企业网之间中断应用连接，根据安全策略(访问控制表)检查应用连接，如果这些应用连接符合安全策略的要求，然后再转发应用连接。——代理服务器
  - 不是过滤报文，而是转换报文 → 网关的基本特征

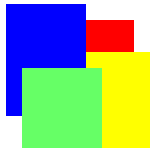




# 网络防火墙部署结构

- 在互联网环境下防火墙一般设置在公共互联网与企业网之间，而企业网中还可以进一步采用防火墙设置成企业外部网和企业内部网。





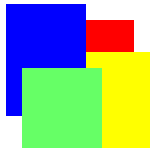
# 各类网络防火墙的功能定义

- 外部防火墙的功能：隔离公共互联网与外部网
  - 外部网：企业放置对外服务器（例如对外网站和邮件服务器）的网络区域。
- 外部防火墙通常是网络层防火墙
- 内部防火墙的功能：隔离外部网与内部网
  - 内部网：企业放置内部服务器和内部网络客户机的网络区域。
- 内部防火墙 = 网络层防火墙 + 应用层防火墙



# 网络防火墙自身安全性

- 防火墙的一个重要特征是自身具有很强的抵御网络攻击的能力。
- 从安全角度看，作为防火墙的路由交换设备或者网关都应该运行简单的、易于控制的、尽量少的程序，关闭不需要的服务。例如应该关闭路由交换设备或者网关上的远地登录服务。
- 应用层防火墙也称为“堡垒主机”。
  - 从用户角度看的应用层防火墙就是一台“主机”
  - 为了使得“堡垒主机”真正成为外部网络攻击无法攻破的“堡垒”，它所运行的操作系统、应用软件都是经过严格筛选和简化的系统和软件。



# 网络层防火墙功能

- 通过禁止(允许)某类源IP地址或者某类目的IP地址，分组过滤器可以限制(允许)来自公共互联网的某些子网访问企业网，或者限制(允许)企业网内某些子网访问公共互联网；
- 通过禁止(允许)某类源端口号或者某类目的端口号，分组过滤器可以限制(允许)来自公共互联网的某类服务请求，或者限制(允许)企业网对公共互联网的某些服务请求。



# 网络层防火墙配置

- S. Bellovin和W. Cheswick建议采用以下三个步骤配置分组过滤器：
- 第一步需要了解被保护网络的**安全策略**，即需要知道什么应该被“**允许**”，什么应该被“**禁止**”，以及缺省情况。
- 第二步需要根据网络安全策略，**寻找**对应的可控制的分组**报头字段**，并且需要**明确**罗列**具体**的安全**控制规则**。
- 第三步按照分组过滤器中ACL要求的格式，**配置**具体罗列的安全**控制规则**。



## 网络层防火墙配置(续1)

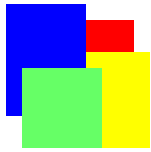
- 假定设置一个分组过滤器，作为公共互联网与企业外部网之间的网络层防火墙。其安全策略是：
  - (1) 只允许公共互联网访问企业外部网中的邮件服务器(210.10.10.36)；
  - (2) 不允许spam.com域名的子网(110.10.160.0)中的站点访问该邮件服务(端口号为25)。
- 要求设置相应的访问控制列表。



## 网络层防火墙配置(续2)

- 该分组过滤器的访问控制列表如下：

动作	源IP地址	源端口号	目的IP地址	目的端口号	说明
禁止	110.10.160.0	*	*	*	禁止spam.com访问
允许	*	*	210.10.10.36	25	允许访问邮件服务器
禁止	*	*	*	*	缺省规则



# 网络层防火墙优点与缺点

- 网络层防火墙优点
  - 成本较低，容易在企业网的边界和内部大范围部署。
- 网络层防火墙缺点
  - 配置较为复杂，需要较为系统的网络知识。
  - 分组过滤器对网络服务的控制能力较弱，设计对网络服务的控制也比较复杂。
  - 缺乏严格的真实性验证机制，网络攻击者可以假冒IP地址和端口号，攻破网络层防火墙。



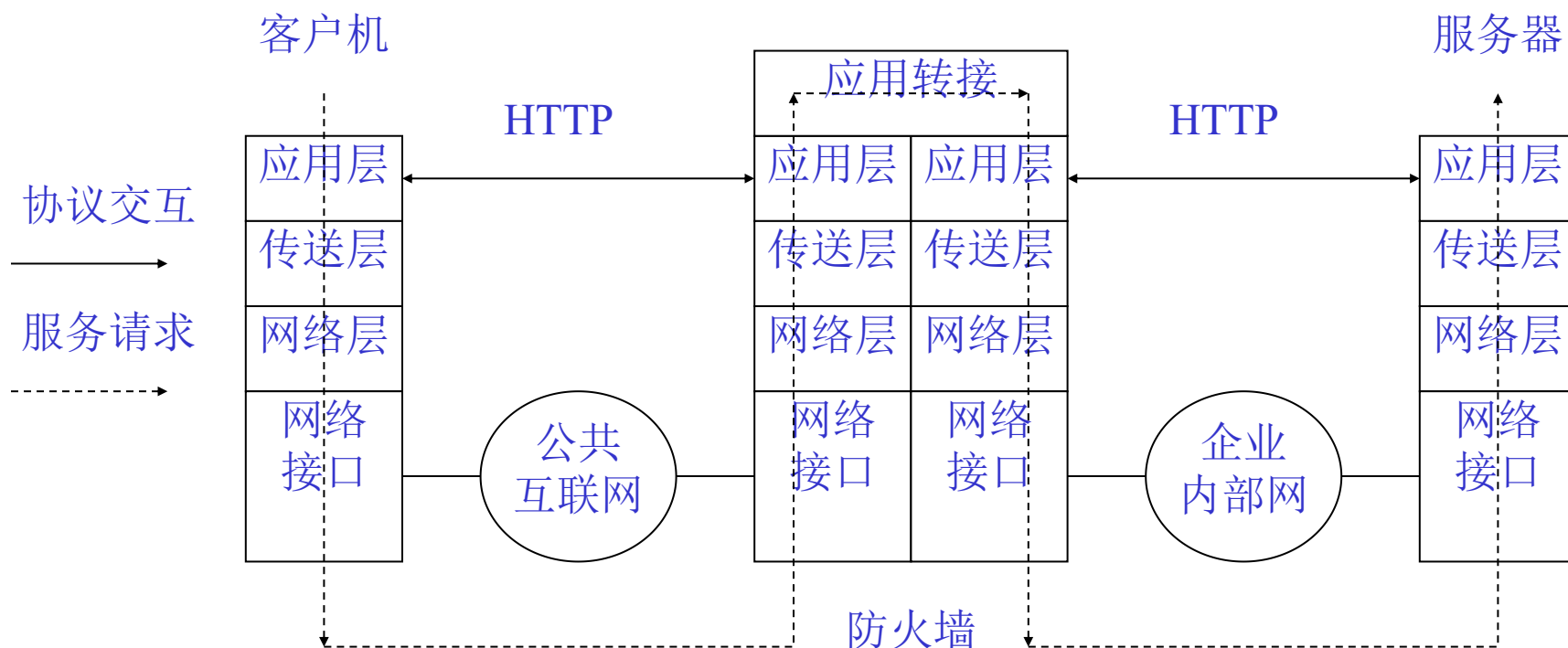
# 应用层防火墙原理

- 应用层防火墙是一种在两个同构网络之间为了安全控制而设置的一个“网关”，用于实现不同安全域的网络应用协议的访问控制和转接。
- 应用层防火墙具有应用代理服务器的功能，即在符合访问控制规则前提下，代理一个网络中的客户端请求另一个网络中服务器的服务。



# 应用层防火墙原理图

- 典型的应用层防火墙处理示意图



应用层防火墙原理图



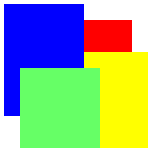
# 应用层防火墙功能

- 应用层防火墙可以根据应用类型进行访问控制，也可以根据网络用户进行访问控制。
- 对于安全控制机制较弱的防火墙，则仅仅识别HTTP/FTP等几个允许的应用协议，如果属于这些应用协议，则防火墙代理客户端向服务器发出服务请求。
- 对于安全控制机制较强的防火墙，则接收到客户端服务请求之后，还要求客户端输入用户名和登录口令，在验证客户端身份之后，才能代理客户端向服务器发出请求。



# 应用层防火墙优点

- 具有配套的真实性验证机制，针对具体的网络应用和网络服务进行安全控制，具有**较强的访问控制能力**。
- 具有较强的自身安全性。由于这种应用层防火墙软件采用**专门设计的应用软件**，仅仅设置了最基本的功能，**软件漏洞较少**。
- 具有较强的应用控制能力，可以在**应用协议、用户身份、应用报文**进行控制，可以在具体网络应用软件中增加特殊的过滤应用层服务请求的功能。



# 应用层防火墙缺点

- 需要为每种应用对应的应用协议设计专门的应用层防火墙软件。
- 对于新出现的网络应用可能造成了较大的障碍（如果缺省控制是“禁止”），有些新出现的网络应用就无法穿越应用层防火墙。



# 网络攻击防御

- 网络攻击定义\*
- 网络攻击分类
- 网络攻击检测系统
- 网络攻击检测方法\*
- 网络蠕虫与检测



# 网络攻击定义

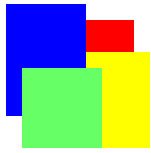
- 危害网络系统及其网络系统中连接的信息系统的安全性的行为统称为“网络攻击”。
- 传统网络攻击对应英文的Network Intrusion，表示对网络系统以及连接在网络系统中的信息系统、计算机系统等的一种侵入行为。现在对应的英文术语是Network Attack，表示一类网络破坏行为。
- “网络侵入”实际上是破坏网络系统的保密性和完整性，“拒绝服务”是破坏网络系统的可用性。
- 网络侵入和拒绝服务可以统称为“网络攻击”。



# 网络攻击历史

- 计算机和网络攻击一览表

年代	对计算机和网络系统的攻击
1980至1985	口令猜测、自我复制恶意代码、口令破解
1985至1990	探测已知缺陷、关闭日志、网络蠕虫、恶意侵入、后门攻击
1990至1995	虚假分组、劫持会话、自动探测扫描、报文嗅探、GUI入侵工具
1995至2000	大规模的拒绝服务攻击、对浏览器的恶意代码攻击、先进扫描技术、基于Windows的远地可控特洛伊代码、电子邮件传播恶意代码、大规模传播特洛伊木马代码、分布攻击工具
2000至今	分布式拒绝服务攻击、大量变种网络蠕虫、基于电子邮件传播恶意代码、基于网页链接传播恶意代码、通过恶意代码获取用户身份信息和账户信息收益、防取证技术、复杂的攻击控制工具



# 近几年网络攻击的变化趋势

- 以下是2014-2015网络攻击变化趋势

1	Malware	↑
2	Web Based Attack	↑
3	Web Application Attack	↑
4	Botnets	↓
5	Denial of Service	↑
6	Physical Damage	↔
7	Insider Threat	↑
8	Phishing	↔
9	Spam	↓
10	Exploits Kits	↑
11	Data Focused Attack	↔
12	Identity Theft	↔
13	Information Leakage	↑
14	Ransomware	↑
15	Cyber Espionage	↑

Notation: ↑ Increasing, ↓ Decreasing, ↔ Same



# 系统渗透类攻击

- 按照Stephen D. Crocker的观点，目前网络攻击可以分成两大类：基于系统渗透的攻击和基于拒绝服务的攻击。
- 基于系统渗透的攻击是攻击者发现被攻击网络系统的(技术或配置)漏洞之后，对网络系统进行的非授权的访问和其他操作。
- 网络系统的漏洞包括技术漏洞（例如缓存区溢出这类软件漏洞）和配置漏洞（例如易于猜测的口令、没有修改默认口令、没有关闭远程登录等）。



# 系统渗透类攻击的防范

- 防范渗透型网络攻击方法：
  - 完整设计坚固的网络体系结构—可以做到；
  - 正确实现这种网络体系结构—可以做到；
  - 严密配置这些网络系统的安全控制策略—可能做到；
  - 严格训练网络用户的安全意识和防范能力—难以做到。
- 迄今为止，防范渗透型网络攻击的效果并不理想。原因在于：无意识地使用具有逻辑漏洞的系统，系统安全配置有误差。
  - 解决方案：人工智能的引入，阿兰·图灵的思路：由机器破解机器的编码→由机器防御机器的攻击！



# 拒绝服务类攻击

- 基于拒绝服务的网络攻击原理：攻击者利用某些手段(植入恶意软件、启动恶意软件的无效网络访问等)，在网络系统中造成大量虚假而正常的网络访问或者网络服务请求，使得网络设备或者网络服务器无法提供正常的服务。
- 拒绝服务(DOS)攻击没有泄露网络系统的信息，也没有渗透到网络系统或应用系统。但拒绝服务攻击已成为互联网上一种最为严重的网络攻击。
  - 注：分布式拒绝服务还需要渗透较多的联网主机



# 拒绝服务类攻击的操作方式

- 目前在互联网上具有较大威胁的网络攻击是**分布式拒绝服务(DDOS)**攻击。DDOS攻击是一种**结合渗透攻击和拒绝服务攻击**的网络攻击方式，它分成两个阶段：
  - 第一个阶段是**渗透阶段**，渗透到尽可能多的计算机系统，在被渗透的系统中设置恶意代码，使得该计算机处于网络攻击者可控制的状态。
  - 第二个阶段是**DDOS攻击阶段**，攻击者向所有已被渗透的主机发出**虚假但正常的**网络或应用访问指令。



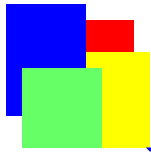
# 传统的网络攻击

- D. Denning在1987年罗列了一些典型的网络攻击行为：
  - 试图闯入：尝试着侵入
  - 假冒者或成功闯入：已经侵入
  - 合法用户的渗透：合法用户访问非授权资源
  - 特洛伊木马：植入的恶意代码，主要的攻击
  - 病毒：具有繁殖、传播和破坏作用的代码
  - 拒绝服务：恶意地过度占用网络资源



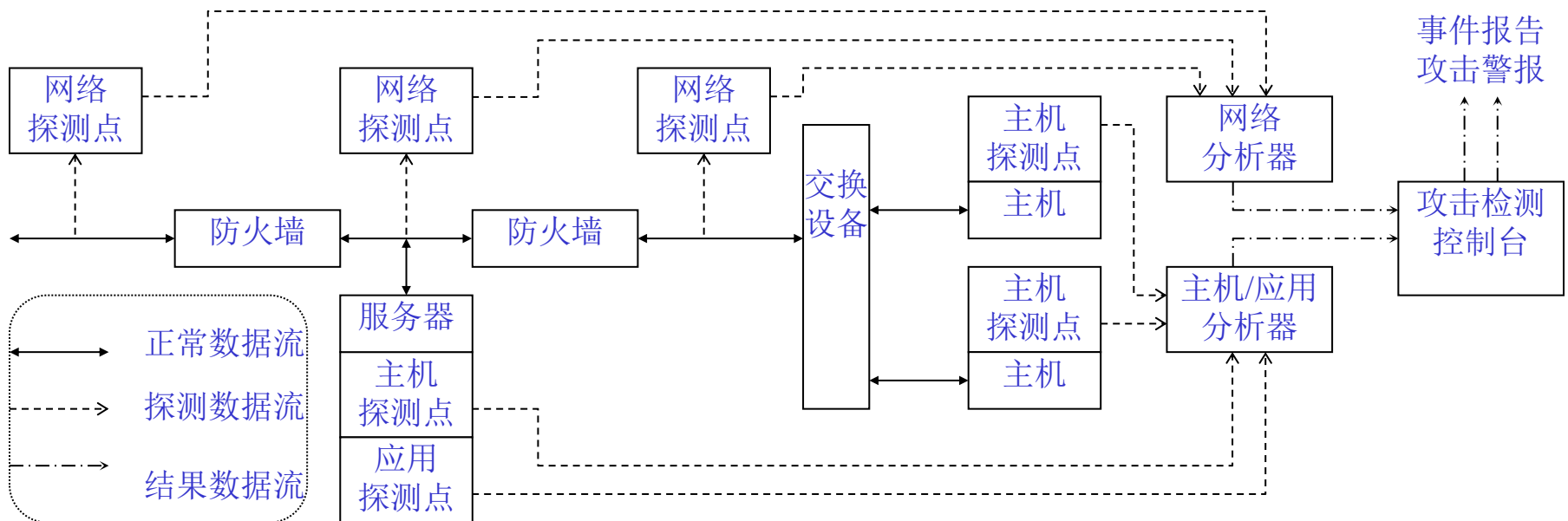
# 现代网络攻击

- **中间人(man in the middle, MITM)攻击**：这是一类在两个或多个网络被害人之间的交互信道上截获、修改报文，或者改变或假冒网络被害人进行的网络攻击。
- **僵尸网(botnet)和暗网(darknet)**：僵尸网是指被网络恶意软件感染、并受恶意软件控制的计算机的在线集合。僵尸网造成DDoS攻击、获取敏感信息，发送垃圾邮件等网络安全威胁。暗网是指其网络服务器或网络操作隐形的网络。暗网可能造成隐形恶意软件传播、DDoS攻击等网络安全威胁。
- **隐形恶意软件(stealth malware)**：这是一类可以躲避检测、在很长时间以自然而睡眠方式传播，用于收集敏感信息，潜伏在关键网络位置发起攻击的恶意软件



# 一个网络攻击检测系统结构图

- J. McHugh等人提出了一个典型的网络攻击检测系统的总体结构。

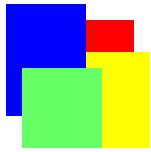


一个典型网络攻击检测系统结构图



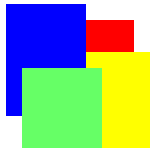
# 网络攻击检测分类

- 按照检测的对象不同，网络攻击检测可以分为基于网络的攻击检测技术和基于主机的攻击检测技术。
- 实际上网络攻击检测技术的本质区别在于检测的方法的不同，正确的网络攻击检测分类方法应该是：基于攻击检测方法的分类。
- 网络攻击检测中，按照攻击检测的方法不同，将网络攻击检测分成特征检测方法和异常检测方法。
- 数据挖掘、机器学习等技术应用于网络攻击检测，可以构成智能网络攻击检测技术(智能的机器防御攻击)。



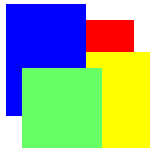
# 网络入侵检测涉及的方面

- **输入数据类型**：网络入侵检测的关键涉及到输入数据的特征，这些数据特征涉及到相关的**数据模型**，例如数据**属性**、数据**类型**、数据**结构**等。数据模型**越能准确描述行为特征**，就越能**准确进行异常检测**。
- **合适的近似度测量**：也是采用合适的**统计分析方法**，其中包括数据的**统计采样和统计分析**方法等。涉及到**统计分析**相关的理论和方法。
- **标记数据**：标记相关的数据是**正常**，还是**异常**。异常行为是**动态变化的**，如果**缺少正常行为模式的数据支撑**（缺少正常行为训练数据），则可能将**正常数据标记为异常**。



# 网络入侵检测涉及的方面-1

- 基于标记数据的分类方法：异常检测可以在监督、半监督和非监督方式下执行，监督方式下具有正常行为和异常行为的训练数据；半监督方式下仅仅具有正常行为的训练数据；非监督方式下没有任何训练数据，机器学习和深度学习从数据中训练(学习)，但需要评判是否满足实时性要求
- 相关的特征识别，异常检测也采用特征识别方法，但这里主要采用基于特征选择的特征识别方法，特征选择方法包括特征子集生成、评估和验证三个步骤。
- 异常报告的输出：异常报告输出通常有两种方式：其一，综合了与简本或特征集的差距和方差、与相邻简本或特征的影响度等得分；其二，综合了非监督分类或聚类方法产生的群的大小和密度等的标签。

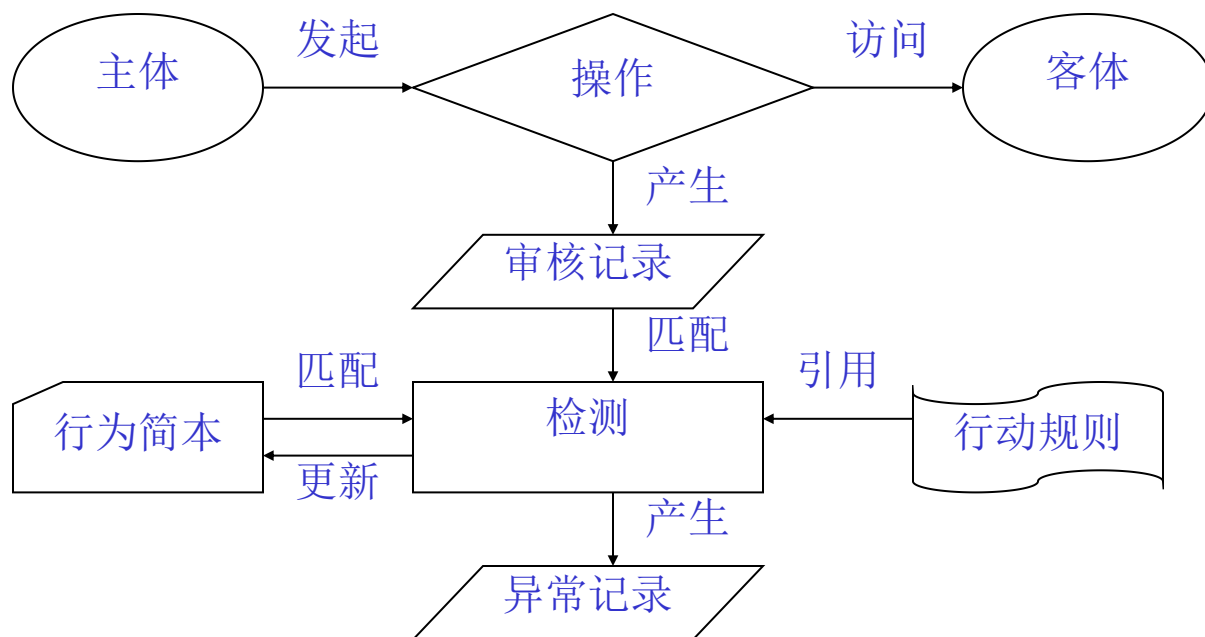


# 网络攻击的异常检测方法

- 基于异常检测的网络攻击检测模型是从网络访问控制模型中自然引申出的一种攻击检测模型，所以，这种模型已经成为一个典型的网络攻击检测模型。
  - 注：基于特征的攻击检测方法常应用网络病毒的检测，通过识别病毒特征之后，采用的过滤方法。这是一类攻击发生之后的防范方法。
- 异常检测模型包括主体、客体、审核记录、行为简本、异常记录和行动规则。

# 传统的异常检测方法的结构图

- 传统的异常检测系统结构图(与访问控制关联)



IDES异常检测系统结构图

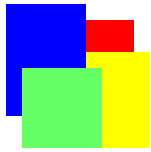


# 异常检测方法的举例

- 审核记录用一个6元组表示：<主体，动作，客体，例外情况，资源使用，时间戳>。
- 网络系统中大部分操作都涉及多个客体，审核记录采集系统必须首先将多客体相关的操作分解成单客体相关的操作，然后再生成审核记录。
- 例如Smith执行的将GAME.EXE文件拷贝到<Library>目录的命令：

COPY GAME.EXE TO <Library>GAME.EXE

- 由于Smith没有“写”<Library>目录的权限，所以，该命令执行失败



# 异常检测方法的举例(续1)

- 针对这项操作需要采用以下3条审核记录表示：  
(Smith, execute, <Library>COPY.EXE, 0, CPU=0002, 11058521678)  
(Smith, read, <Smith>GAME.EXE, 0, RECORD=0, 11058521679)  
(Smith, write, <Library>GAME.EXE, write-viol, RECORD=0, 11058521680)
- 通过分析主体对客体的异常访问，发现正在实施或已经实施的网络安全攻击；通过分析主体对客体访问的例外情况分析，发现潜在的网络攻击——数据挖掘



# 攻击检测与审核记录

- 如果使用基于网络的攻击检测系统，则审核记录由网络探测器产生；如果使用基于主机的攻击检测系统，则审核记录由主机探测器产生。
- 审核记录的产生时间决定了攻击检测的能力。
  - 如果审核记录是在操作开始时就生成，则攻击检测系统可以利用审核记录，检测潜在的 attack 或者正在进行的攻击；
  - 如果审核记录是在操作完成后产生，则攻击检测系统可以利用审核记录，检测已经实施的、或已经完成的攻击。



# 网络蠕虫\*

- “网络蠕虫”是一类自我复制和自我传播的、无需人工介入传播的恶意程序或者恶意代码。蠕虫的表现形式与网络上的移动代码有些类似，
  - 移动代码是一类网络上自我传播和自我执行的程序或者代码。移动代码是为了方便网络管理和配置而设计和部署的一类可以在网络上自动传播的合法代码，这类移动代码一般称为“移动智能体”或“移动代理”。
- 网络蠕虫依然是互联网上的头号安全威胁，它曾在国际上造成了上百亿美元的经济损失，防范和控制网络蠕虫已经成为网络安全研究的头等大事——现在的防火墙可以防范网络蠕虫。
  - 网络蠕虫一般通过被攻击的网站或虚假网站、电子邮件、共享文件而传播。



# 莫里斯蠕虫

- 第一个在互联网上造成重大影响的网络蠕虫是莫里斯(Morris)蠕虫，该网络蠕虫爆发于1988年11月2日晚。莫里斯蠕虫事件使得互联网结束了乌托邦式信任阶段，进入到了异构的、危险的、全球范围的互联网新时期。从那时开始起，人们开始研究互联网防火墙技术、互联网病毒防范技术。
- 恶意代码是一种有意设计的程序，用于执行非授权的，通常是有害的或不受欢迎的动作。“病毒”就是一种恶意代码，它是一种具有自我繁殖能力的恶意代码。而“蠕虫”也是一种恶意代码，它不仅具有自我繁殖能力，而且还可以在网络上进行传播的一种恶意代码。



# 网络蠕虫相关概念

- “特洛伊木马”是一种看似有用的计算机程序，但是，它隐藏了潜在有害的功能。“特洛伊木马”也是一种恶意代码，它与“病毒”和“蠕虫”的区别是：没有自我复制的能力，即没有自我繁殖的能力。
- “远地访问特洛伊”是一种“特洛伊木马”，一旦被执行，它允许非授权的用户远地访问控制被它攻破的系统。
- “后门”，有时也称为“陷门”，是程序设计者在程序中设置的一项功能，它允许设计者完全的或者部分的非授权访问执行该程序的系统。



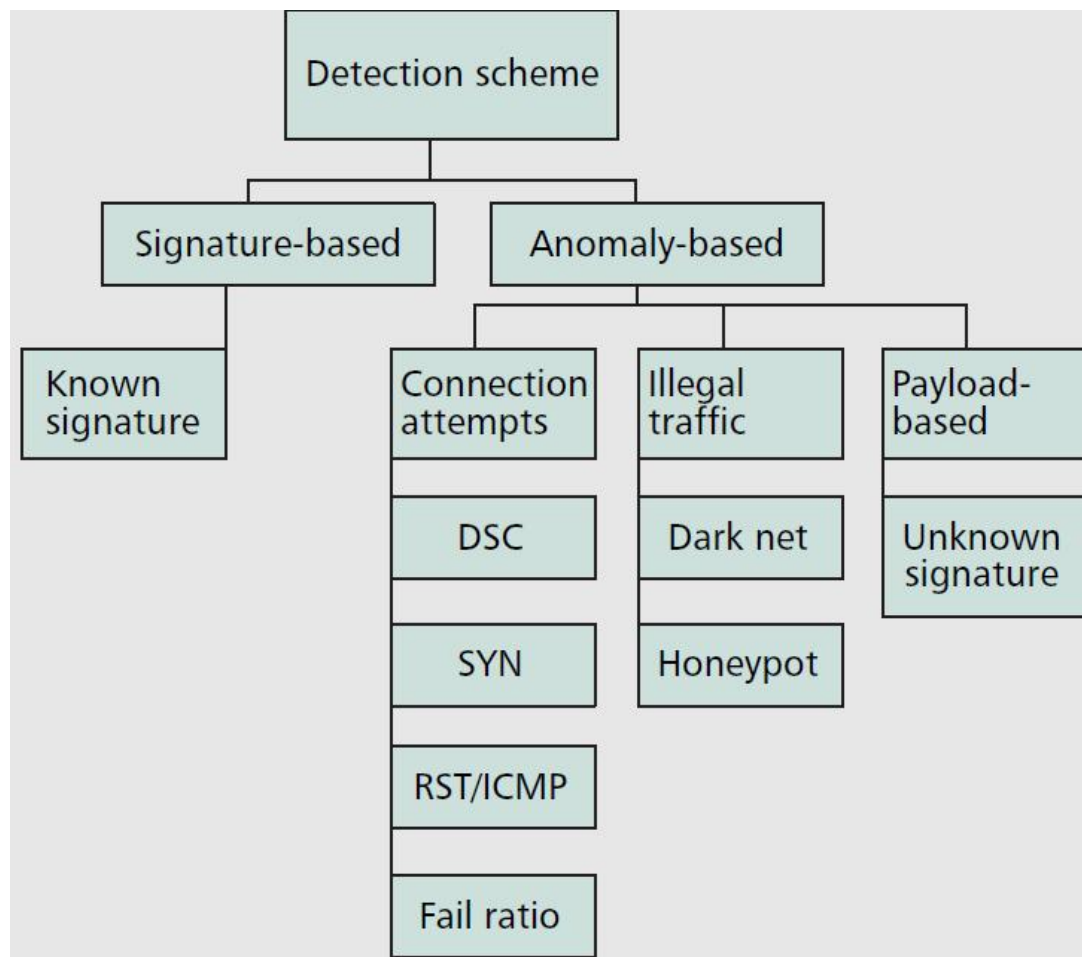
# 网络蠕虫分类

- D. Kienzle和M. Elder通过对蠕虫特征的分析，将蠕虫分成以下3种类型[22]：电子邮件蠕虫，Windows文件共享蠕虫和传统蠕虫。
- 电子邮件蠕虫是利用电子邮件机制在网络上传播的一类蠕虫；
- Windows文件共享蠕虫是利用Windows工作组文件目录共享机制在网络上传播的一类蠕虫；
- 传统蠕虫是采用传统莫里斯蠕虫设计技术，利用TCP/IP协议的连接在网络上传播的一类蠕虫。



# 互联网蠕虫检测方法

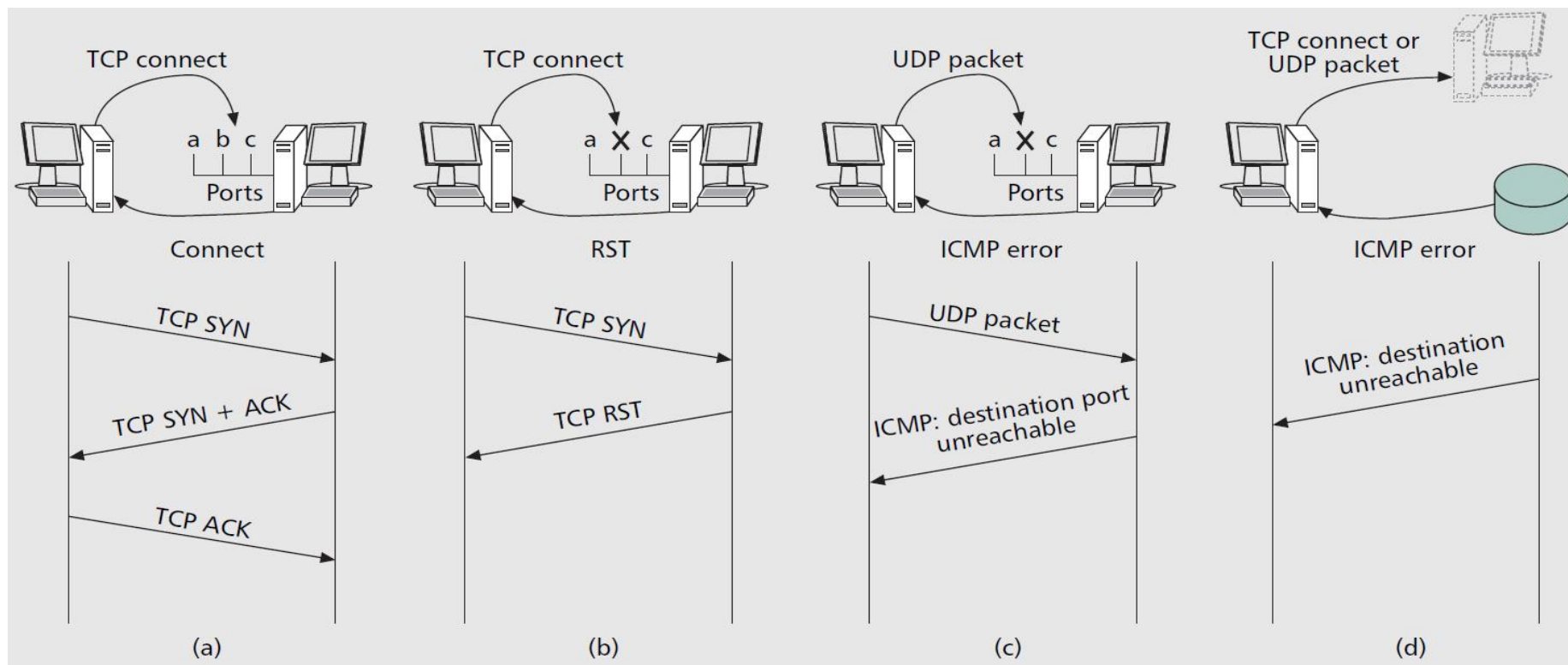
- 互联网蠕虫的检测方法分成两类：
  - 特征检测方法
  - 异常检测方法，包括连接异常检测方法；流量异常检测方法；分组内容异常检测方法

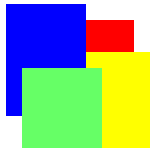




# 连接异常的蠕虫检测

- 连接尝试过程中的异常现象：a) 成功TCP连接次数过多；b) TCP目的端口关闭；c) UDP目的端口关闭；d) 目的IP地址不存在。可以用于检测蠕虫的扫描。



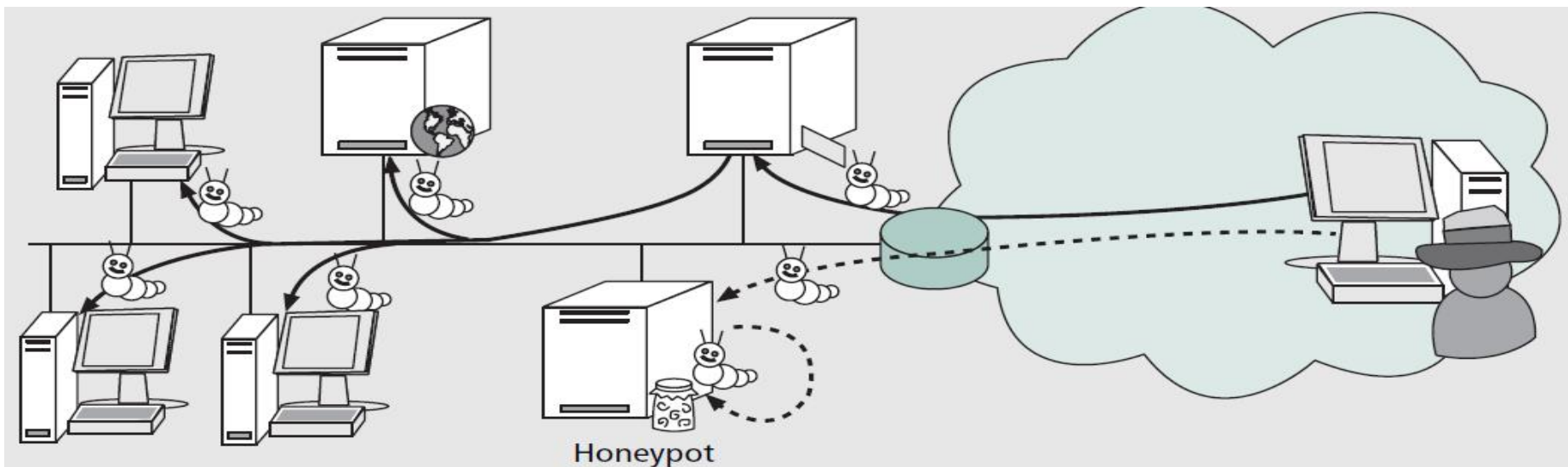


# 连接异常的蠕虫检测说明

- 从某个主机发出的TCP的连接请求报文(SYN)数在某个时间段内超过某个阈值，则可以认为这个主机正在扫描（a）。
- 按照TCP/IP协议，如果目的主机的IP地址不存在，则返回ICMP主机不可达的分组（d）。
- 如果TCP连接的目的端口关闭，则返回TCP RST分组（b）；
- 如果UDP报文中的目的端口关闭，则返回ICMP目的不可达的分组（c）。

# 蜜罐检测和隔离方法

- 蜜罐是一个网络中不提供任何真实服务，但却是一个易攻击(例如对于公开的系统漏洞没有打补丁)的主机或应用系统，这是一类安全资源，其价值在于被探测、攻击和捕获。
- 由于蜜罐是没有任何正常的服务请求，任何对于蜜罐的服务请求都是异常的。蜜罐获得的数据较少，但价值很高。可以检测盲扫描、攻击名单扫描、拓扑类蠕虫，但无法检测被动蠕虫。





# 基于分组内容的蠕虫检测

- 基于分组传递信息的检测可以检测对于网络协议栈的漏洞攻击、或者对于主机的服务漏洞的攻击，这是基于网络的攻击检测。
- 而针对应用漏洞的攻击检测则无法基于分组传递行为，必须基于分组内容进行检测。因为这类攻击的连接都已经正常建立，分组传递不再有任何异常。
- 基于分组内容的检测包括：对于正常分组内容长度的统计分析；分组的特定应用（与分组的端口号相关）与分组内容长度关联的统计分析等。



# 蠕虫的隔离

- 降速：采用反馈回路延迟可疑的流量，在不同网络功能层采取限速技术等。由于蠕虫传播速度极快，这些降速机制必须是自动的，最好是能够自动阻隔。
- 阻隔：当发现类似蠕虫行为，则蠕虫行为发起的网络源（包括主机源、网关源等）必须与其他网络隔离，避免更多的网络或主机被感染。
- 诱捕：采用在蜜罐主机中部署多个虚拟机和虚拟地址、端口的方式，使得进入蜜罐的蠕虫无法再向外传播。



# 蠕虫的阻隔方法

- 蠕虫的阻隔与降速是联动使用的：
  - 当检测到蠕虫行为的警告达到第一级阈值，则系统首先是降速，避免误报；
  - 当情况恶化并且达到了第二级警告阈值，则采用阻隔。
- 阻隔包括两类方法：
  - 基于分组内容的阻隔：丢弃携带蠕虫代码特征内容的分组；丢弃扫描到关闭端口而报错的分组。
  - 基于分组地址的阻隔：对于发出蠕虫的主机地址进行阻隔，通常在网关上设立黑名单。



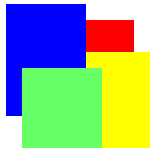
# 网络安全加固

- 安全IP概述\*
- 真实性验证报头(AH)协议
- 封装安全报体(ESP)协议
- 互联网安全关联与密钥管理协议(ISAKMP) \*
- 互联网密钥交换(IKE)协议\*
- SSL记录协议
- SSL握手协议



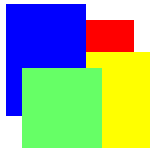
# 安全IP概述

- 安全IP(IPsec)在互联网协议(IP)层提供安全服务，也就是在网络层提供安全服务。
- 安全IP提供的安全服务包括：访问控制、数据传递的完整性验证、数据源真实性验证、防范重播分组攻击、数据保密传递、以及有限的的数据传递信息保密。
- 这里“有限的数据传递信息保密”是指不泄漏IP分组真实的源地址、目的地址、端口号等协议控制信息。



# 安全IP的总体组成

- IPsec总体上包括4个组成部分：安全协议，安全关联，密钥管理，以及真实性验证算法与加密算法。
- 为了利用IPsec在IP层提供安全服务，必须选择安全协议、选择安全协议中采用的真实性验证算法或者加密算法，协商真实性验证算法或加密算法采用的密钥，最终建立需要进行IPsec通信的IP结点之间的安全关联。



# 安全IP中定义的安全协议

- IPsec目前只提供两种安全协议：真实性验证报头(AH)协议和封装安全报体(ESP)协议。
- AH协议主要提供的IP层安全服务包括：访问控制、数据传递的完整性验证、数据源真实性验证和防范重播分组攻击。
- ESP协议不仅可以提供AH协议提供的真实性验证类安全服务，还可以提供数据保密传递和有限的数据传递信息保密等功能。



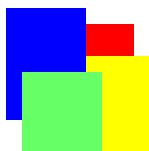
# 安全关联

- 安全关联(SA)概念是安全IP的基础，AH和ESP都需要使用SA，而互联网密钥交换(IKE)协议的主要功能是建立和维护安全关联。
- 安全关联是两个或者多个实体之间描述如何使用安全服务进行安全通信的一种连接关系，这种关系采用这些实体之间作为合同的一组信息表示。
  - 这组信息是在这些实体之间协商和共享的，有时这组信息本身就称为一个“安全关联”。
- 安全关联是一个单工(单向)连接，它为在该连接上传递的报文提供安全服务。
- SA可以利用AH协议或者ESP协议提供安全服务，但是，一个SA不能同时使用AH和ESP协议。



# 安全关联的标识

- 一个SA可以由三元组（安全参数索引, IP目的地址, 安全协议标识）唯一标识,
- 安全参数索引(SPI)指向存放该SA已经协商完成的安全参数的数据项,
- IP目的地址指向该SA数据接收方的主机或安全网关,
- 安全协议标识表示SA采用的安全协议是AH, 还是ESP。

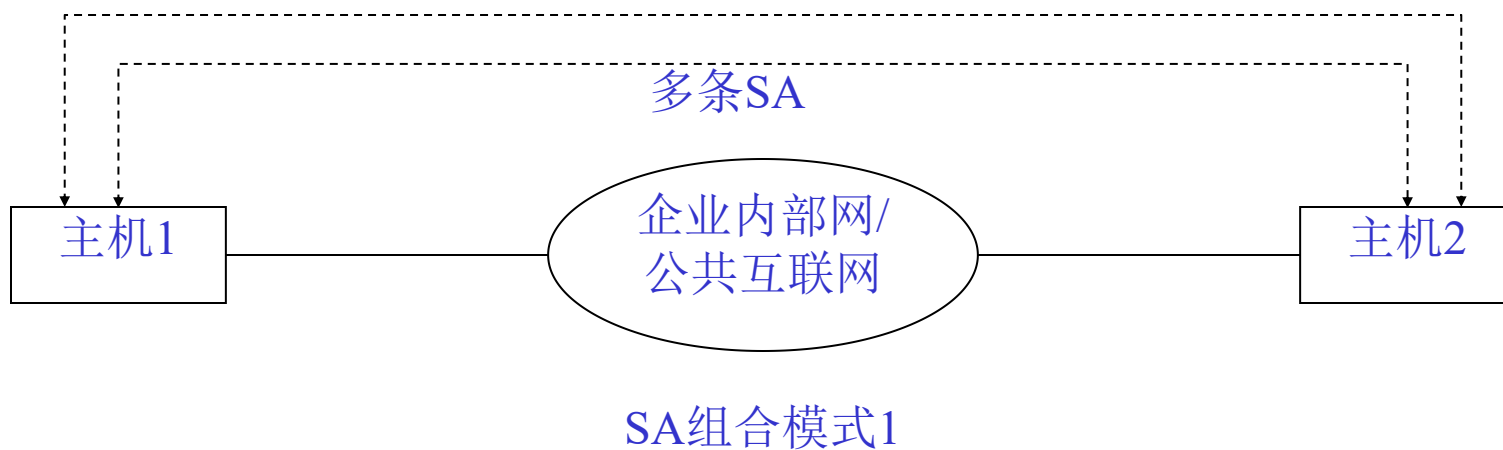


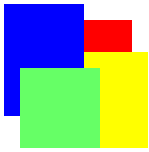
# 安全关联的模式

- SA可以分成两种模式：一种是**传送模式**，另一种是**隧道模式**。
- **传送模式的SA**是在两个主机之间建立的SA，它仅仅保护IP层之上的报文传递。例如，传送模式的SA可以采用ESP协议对传送层报文进行加密传递。
- **隧道模式的SA**是将安全关联应用于IP隧道中。

# SA组合模式1

- 在两台IP网络主机之间可以建立多条SA(如图6.2所示), SA可以是传送模式或者隧道模式。这里的主机1和主机2都支持IPsec相关协议。





## SA组合模式1 (续)

- 在主机1和主机2之间传递的分组报头可以采用以下任意一种形式：

传送模式1: [IP1][AH][Upper]

传送模式2: [IP1][ESP][Upper]

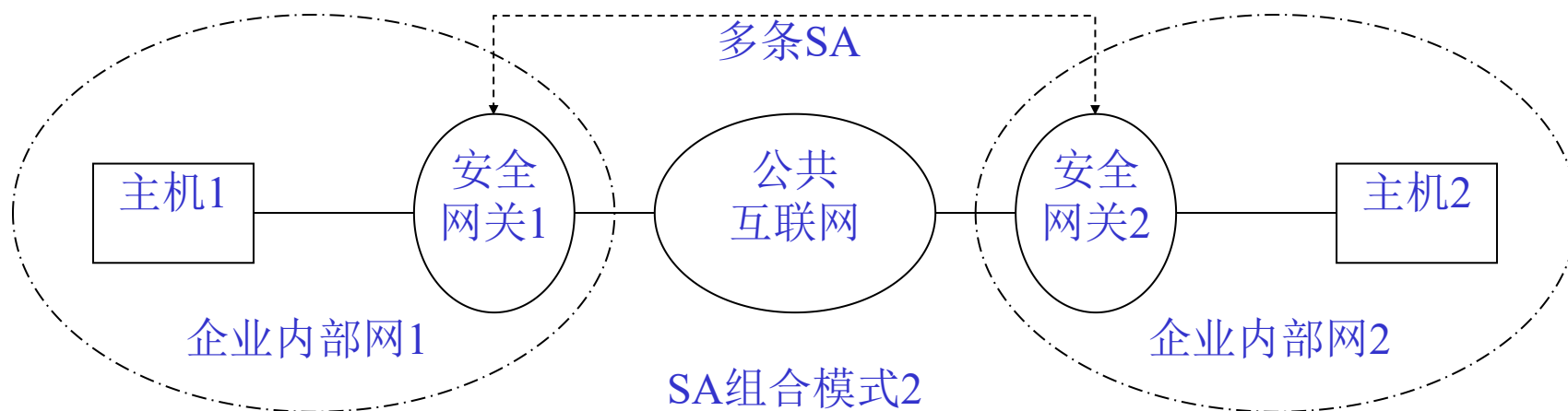
传送模式3: [IP1][AH][ESP][Upper]

隧道模式1: [IP2][AH][IP1][Upper]

隧道模式2: [IP2][ESP][IP1][Upper]

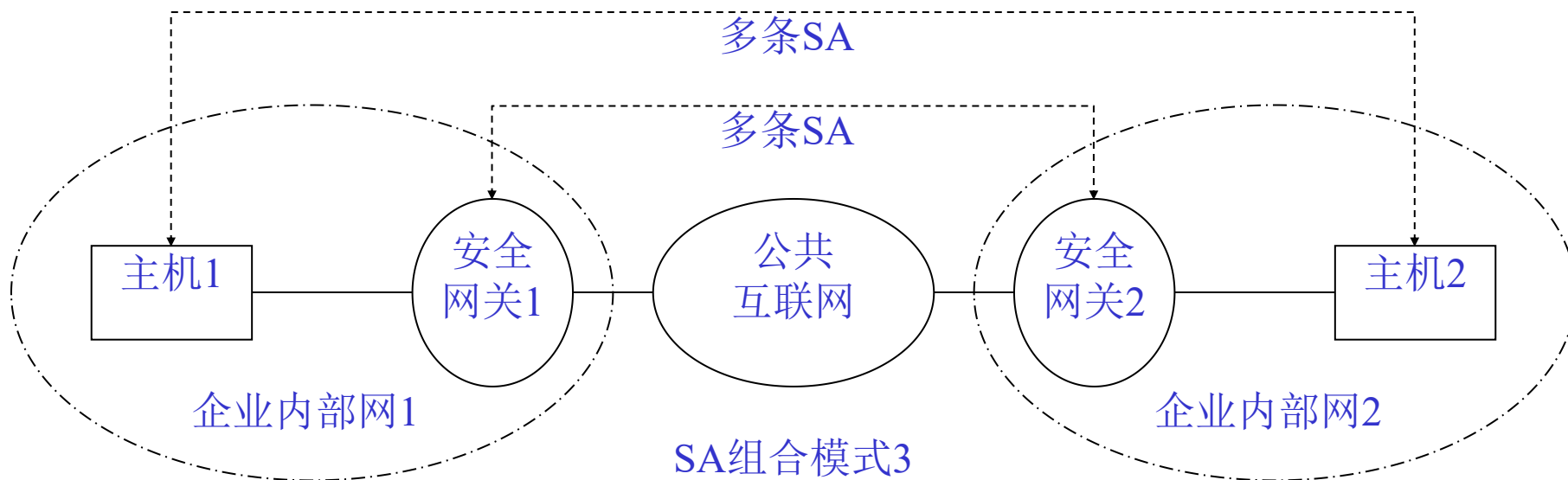
## SA组合模式2

- 这是简单的虚拟专用网(VPN)支持模式，两个企业内部网通过两个安全网关建立的多条SA实现跨越公共互联网相互连接。



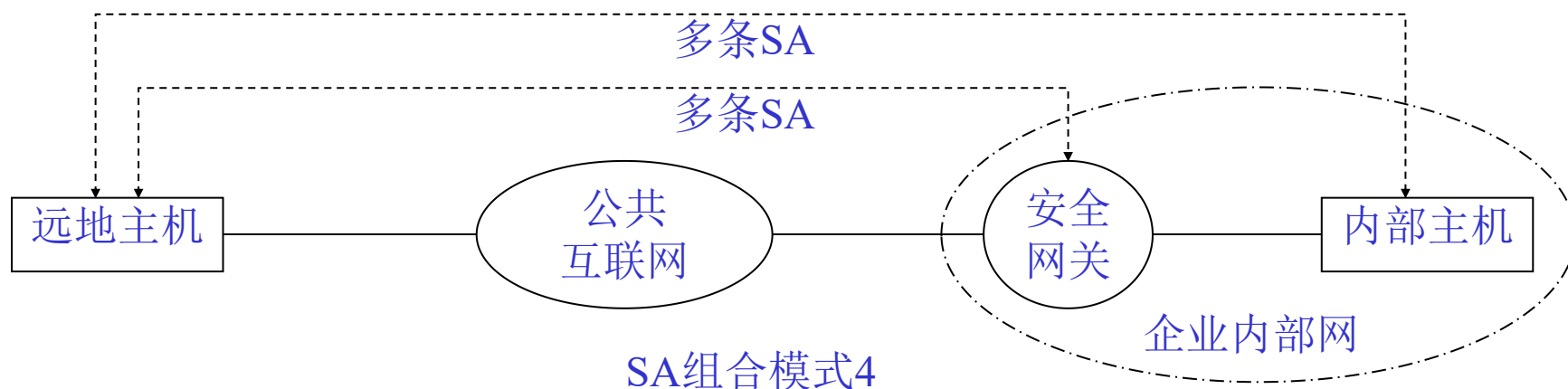
# SA组合模式3

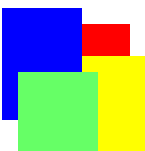
- 这是SA组合模式1与SA组合模式2的结合，在SA组合模式2的基础上增加了IP报文的发送主机和接收主机之间的多条SA。



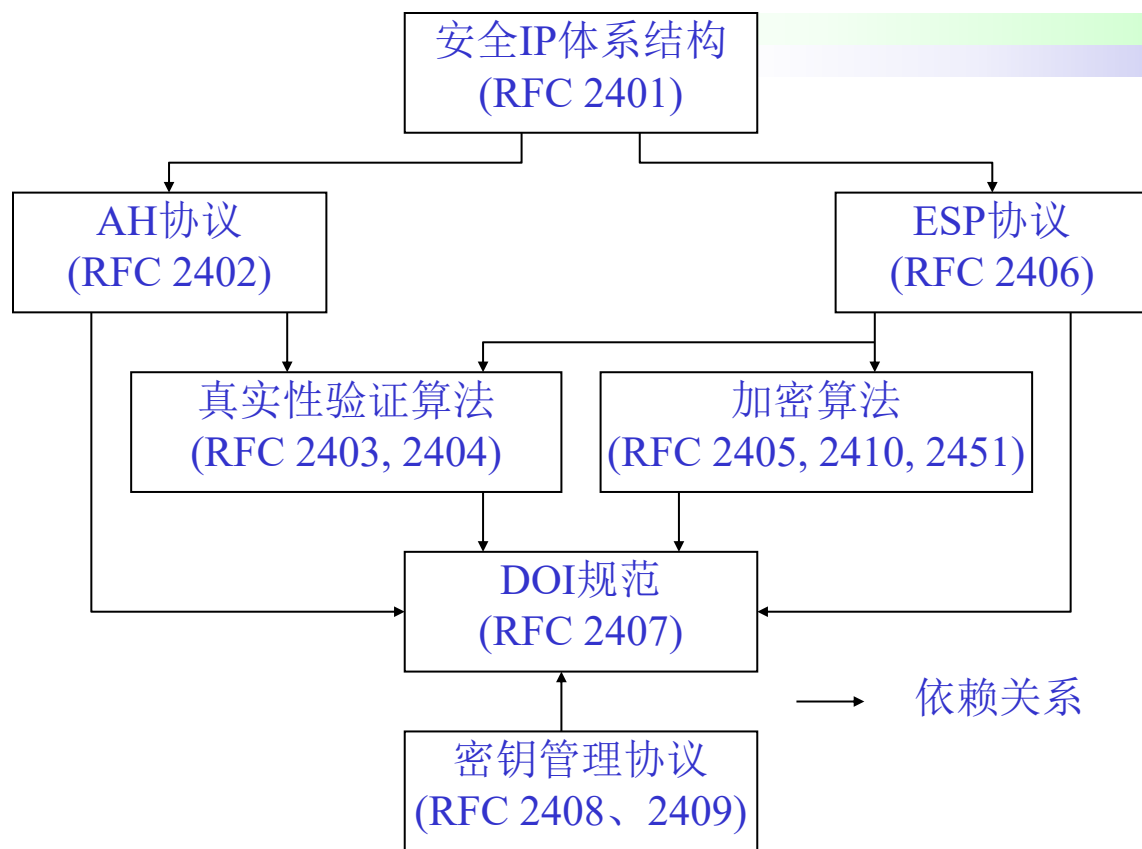
# SA组合模式4

- 远地主机首先通过公共互联网与安全网关建立SA连接，然后，再与企业内部网中某个主机建立SA连接。



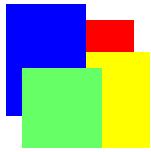


# IPsec协议簇



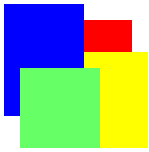
安全IP协议簇组成结构图

- IPsec由一组IETF定义的IPsec技术标准描述，这组协议称为IPsec协议簇。



# 真实性验证报头(AH)协议

- 真实性验证报头(AH)协议IPsec中定义的两个安全协议之一。
- AH主要对IP报文提供无连接传递的完整性验证以及对数据源的真实性验证，它也可以提供防范IP报文重播攻击的功能。
- AH协议真实性验证的范围包括尽可能多的IP报头的内容，以及IP报文携带的数据。



# AH协议报文格式

- AH协议报文包括：下个报头、报体长度、预留、安全参数索引(SPI)、顺序号和真实性验证数据，这6个AH报文必须的字段。

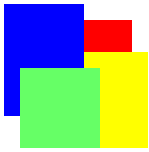
0	7 8	15 16	31比特
下个报头	报体长度	预留	
安全参数索引(SPI)			
顺序编号			
真实性验证数据(长度可变)			

AH协议报头格式



# AH协议的处理

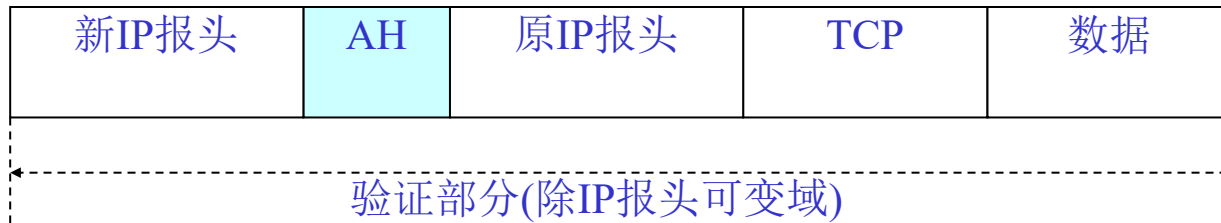
- AH协议报文实际上就是一个报头。AH协议编号为51，AH报文将插在IP报头与IP报体之间。
- 与SA的使用模式一样，AH也有2种使用模式：传送模式和隧道模式。在两种模式中，AH报文的位置有所不同。



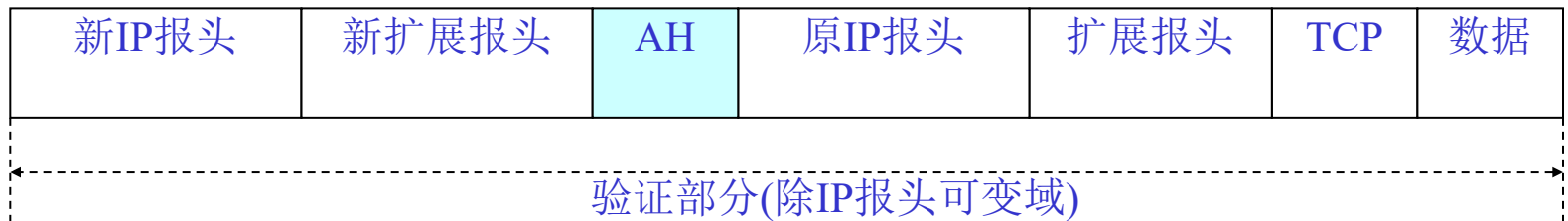
# AH协议的处理(续2)

- 隧道模式中，AH报文的封装格式：

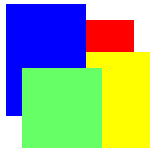
IPv4



IPv6

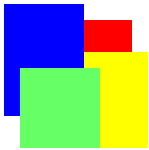


隧道模式中AH报文的封装格式



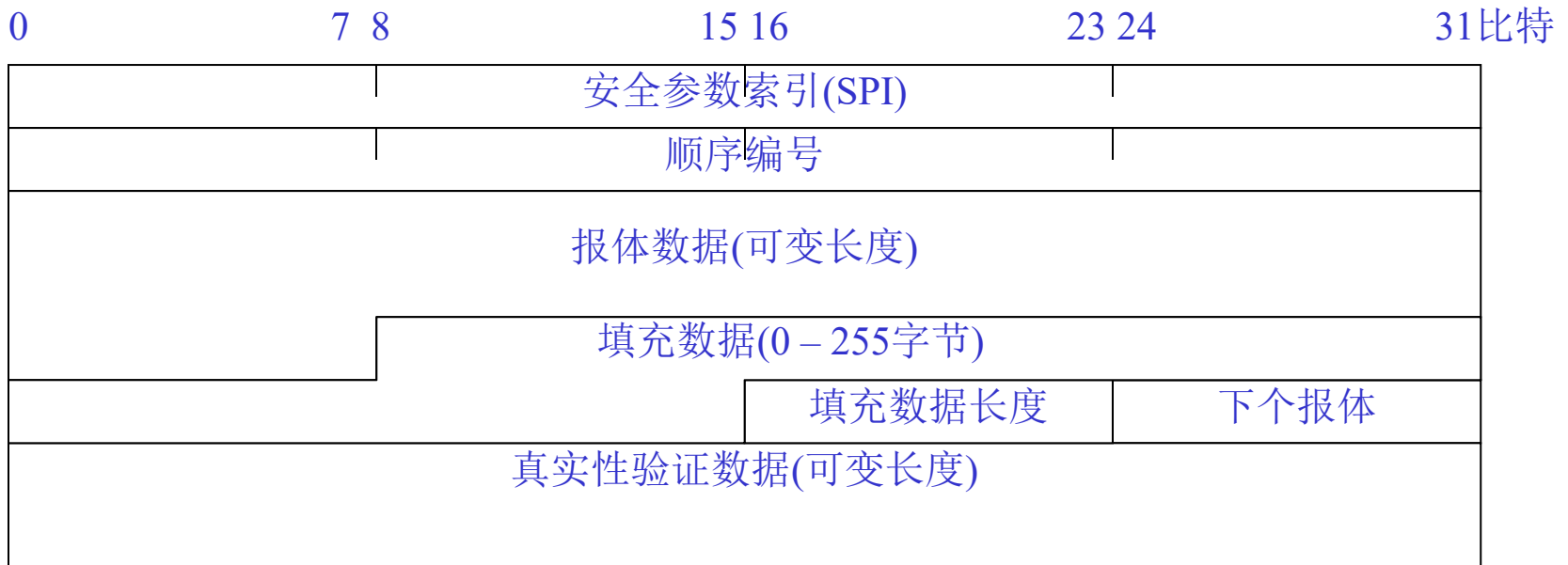
# 封装安全报体(ESP)协议

- 封装安全报体(ESP)是安全IP技术中定义的两个安全协议之一。
- ESP主要用于对IP报文提供保密传递、无连接传递的完整性验证以及对数据源的真实性的验证。ESP也可以提供防范IP报文重播攻击的功能，以及有限度的通信流保密性。
- ESP主要提供对IP报文加密传输的功能，它是专门为对称密钥加密算法设计的安全协议。



# ESP协议报文格式

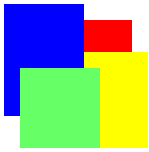
- ESP报文包括安全参数索引(SPI)、顺序编号、报体数据、填充数据、填充数据长度、下个报头、以及真实性验证数据。





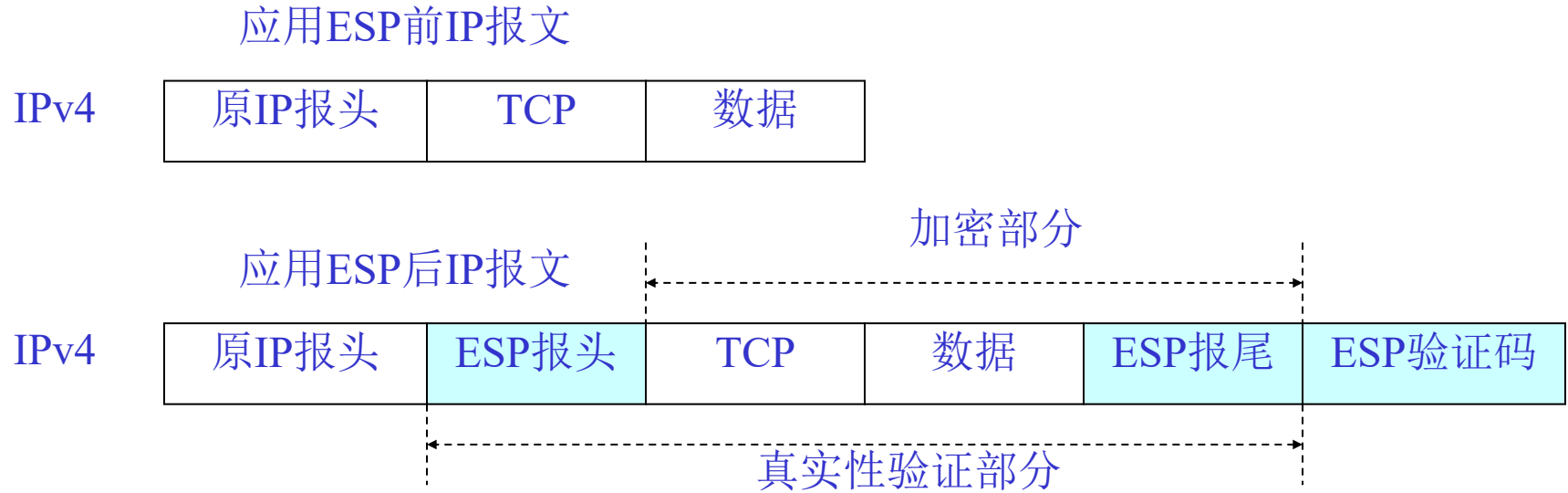
# ESP协议的处理

- ESP协议编号是50，封装ESP协议的报文需要在“下个报头”字段中设置“50”。
- ESP协议不同于AH协议，它是将需要保护的IP报文(隧道模式)或者IP报文传递的报体(传送模式)封装在自己的“报体数据”字段中。
- 对于IP报文应用ESP协议时，需要考虑ESP报头和报尾在原来IP报文中的位置。

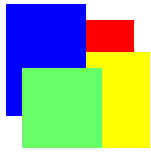


## ESP协议的处理(续1)

- 与SA的使用模式一样，ESP也有2种使用模式：  
传送模式和隧道模式。
- 传送模式中，ESP报文对IPv4的封装格式如下：

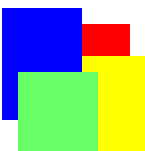


ESP报文的IPv4传送模式封装格式



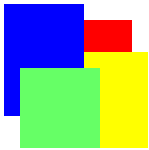
# 互联网安全关联与密钥管理协议

- 互联网安全关联与密钥管理协议(英文缩写ISAKMP)集成了真实性验证、密钥管理和安全关联的网络安全概念，为互联网上政府、商业和个人通信创建一个安全的环境。
- ISAKMP虽然上在IPsec协议簇中定义的一个协议，但是，它试图适用于互联网的所有协议层。



# ISAKMP功能

- ISAKMP定义了
  - 一个标准的ISAKMP报头格式、
  - 一组用于安全关联的创建、修改和删除的标准报体格式、以及
  - 一个标准的安全关联创建的两阶段过程。
- ISAKMP对于设计和实现互联网环境下的密钥管理协议具有很好的指导作用。

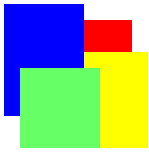


# ISAKMP报头格式

- ISAKMP报头包含了维护协议状态、处理报体、以及防范拒绝服务和重播攻击所需要的信息。

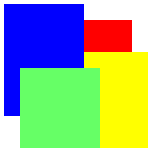


ISAKMP报头格式



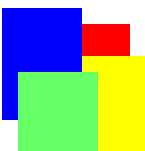
# SA创建的两阶段过程

- ISAKMP需要通过两个阶段才能建立针对某个具体安全协议的安全关联。
- 第一个阶段是建立ISAKMP自身的安全关联，称为ISAKMP安全关联；
- 第二个阶段再建立针对某个具体安全协议的安全关联，称为协议安全关联。
- 两阶段协商过程保证了ISAKMP独立于任何具体的安全协议，并且可以适用于任何安全协议（不仅包括AH和ESP协议）的协商过程。



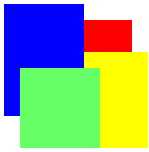
# ISAKMP交换类型

- 无论是第一阶段协商，还是第二阶段协商，都采用ISAKMP定义的交换机制，或者采用密钥交换协议定义的交换机制。
- 目前ISAKMP定义了5种缺省的交换类型：基本型、标识保护型、真实性验证型、自信型、以及消息型交换，用于建立和修改安全关联。
- 不同ISAKMP交换类型之间的主要区别是交换报文的次序，以及在单个报文中报体的次序。



# ISAKMP说明

- ISAKMP协议中定义的交换类型实际上并没有构成一个完整的协议，为了真正建立和修改安全关联，还需要定义较为完整的真实性验证和密钥交换协议。
- 互联网密钥交换(IKE)协议就是一种较为完整的安全关联建立和密钥协商协议。
- IKE协议还是采用了ISAKMP定义的ISAKMP报头和报体格式，以及ISAKMP两阶段协商框架结构。



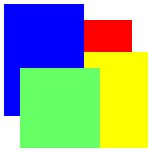
# 互联网密钥交换协议

- 互联网密钥交换(英文缩写IKE)协议是一种混合协议，它在保护模式下协商和提供安全关联所需要的经过真实性验证的密钥资料。
- **IKE**定义的交互模式可以应真实性验证AKMP定义的第一阶段和第二阶段的协商，但是这些交互模式只能应用于ISAKMP定义的两种交互阶段之一。



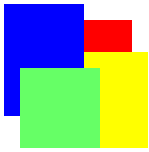
# IKE交互模式

- IKE定义的“主模式”和“自信模式”用于完成第一阶段ISAKMP安全关联的交互，并且只能用于第一阶段交互。
- 第二阶段是代表某类安全服务协商，例如IPsec安全服务，进行的交互，目的是建立某类安全协议要求的安全关联。第二阶段交互需要协商密钥资料和参数。IKE定义的“快速模式”可以完成第二阶段的交互，并且只能应用于第二阶段的交互。



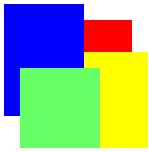
## IKE交互模式(续1)

- IKE定义了应用于第一阶段的两种基本的真实性验证密钥交换模式：主模式和自信模式。
- 这两种模式都是从短期的Diffie-Hellman交换中生成经过真实性验证的密钥资料。主模式是必须实现的，而自信模式是应该实现的。
- 主模式是ISAKMP标识保护型交换的一种实例。自信模式是ISAKMP自信型交换的一种实例。



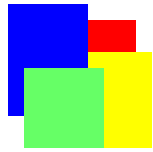
## IKE交换模式(续2)

- IKE定义了应用于第二阶段的密钥交换模式：**快速模式**。
- **快速模式**也是必须实现的一种机制，用于生成最新的密钥资料、协商非ISAKMP的安全服务。
- **快速模式**在ISAKMP中没有对应的交换类型，它们是IKE中定义的密钥交换的操作模式。



## IKE交换模式(续3)

- IKE定义的“**新组模式**”既不属于第一阶段，也不属于第二阶段。它是在第一阶段之后建立一个用于未来协商的新组。所以，它属于ISAKMP第一阶段协商之后。
- 加密算法、哈希算法、真实性验证算法、Diffie-Hellman密钥生成数据都是**IKE使用的算法和参数**，它们是在ISAKMP安全关联中协商的。
- 这些算法和参数仅应用于ISAKMP安全关联中，不一定适用于ISAKMP为其他密钥协商协议建立的安全关联。



# IKE的交换协议

- IKE在交换模式的基础上具体定义了基于真实性验证的密钥交换协议。
- 利用主模式和自信模式，IKE可以采用4种不同的真实性验证方法：数字签名、公钥加密真实性验证、改进的公钥加密真实性验证、以及预共享密钥。
- IKE定义的交换协议完全采用ISAKMP定义的报体格式、参数编码、报文超时和重发、以及消息报文。



# IKE协议符号定义

- **A**表示发起方，**B**表示响应方。注意在RFC 2409中，**I**表示发起方，**R**表示响应方。为了保证全书符号尽量统一，这里采用本书在“真实性验证协议”章节中的标记符号。
- **HDR**表示ISAKMP报头，**HDR\***表示ISAKMP报头后面的报体被加密。
- **SA**表示ISAKMP定义的SA报体，SA报体可以带有一个或者多个建议报体。**发起方**可以提供**多个建议报体**，而**响应方**只回复一个**被选中的建议报体**。在IKE具体交互中，为了简化交互报文的描述，没有明确表示“建议”报体。



## IKE协议符号定义(续1)

- $\langle P \rangle$ 表示报体P中的内容。例如 $\langle SA \rangle$ 表示SA报体的内容。
- $CKY_A$ 和 $CKY_B$ 分别表示在ISAKMP报头中的发起方和响应方“甜点”。
- $G^{XA}$ 和 $G^{XB}$ 分别表示发起方和响应方的Diffie-Hellman的公共值。
- $G^{XY}$ 表示Diffie-Hellman算法生成的共享密钥。
- $KE$ 表示密钥交换报体，其中包含了Diffie-Hellman交换中需要的公共信息。
- $N_A$ 和 $N_B$ 分别表示ISAKMP发起方和响应方一次性数报体。



## IKE协议符号定义(续2)

- $ID_{A1}$ 和 $ID_{B1}$ 分别表示ISAKMP第一阶段协商过程中ISAKMP安全关联的发起方和响应方的标识报体。
- $ID_{A2}$ 和 $ID_{B2}$ 分别表示第二阶段协商过程中，用户或者应用安全关联的发起方和响应方的标识报体。
- SIG表示数字签名报体。
- CERT表示证书报体。
- HASH表示哈希报体， $HASH_A$ 和 $HASH_B$ 分别表示发起方和响应方的哈希报体。
- $PRF(K, M)$ 是基于密钥K对报文M生成的伪随机函数，通常是基于密钥的哈希函数，用于产生确定性的伪随机输出。该函数可以用于生成密钥和真实性验证。



## IKE协议符号定义(续3)

- **SKEYID**表示从只有参与方知道的保密资料中导出的字符串。
- **SKEYID<sub>E</sub>**表示ISAKMP安全关联用于保护报文保密性的密钥资料。
- **SKEYID<sub>A</sub>**表示ISAKMP安全关联用于报文真实性验证的密钥资料。
- **SKEYID<sub>D</sub>**表示用于导出非ISAKMP安全关联密钥的密钥资料。

# IKE交换1：数字签名真实性验证的第一阶段协商

- 原理：该交换协议通过签名双方都可以获得的哈希值进行真实性验证。采用数字签名真实性验证的主模式交换过程如下：

M1: A  $\rightarrow$  B: HDR, SA

M2: B  $\rightarrow$  A: HDR, SA

M3: A  $\rightarrow$  B: HDR, KE,  $N_A$

M4: B  $\rightarrow$  A: HDR, KE,  $N_B$

M5: A  $\rightarrow$  B: HDR\*,  $ID_{A1}$ , [CERT,]  $SIG_A$

M6: B  $\rightarrow$  A: HDR\*,  $ID_{B1}$ , [CERT,]  $SIG_B$



# IKE交换1(续1)

- M1中的SA包括了多个“建议报体”，而M2中的SA只包括一个B选择的“建议报体”。
- M3和M4中的KE报体包括了Diffie-Hellman密钥生成算法的公共值 $G^{XA}$ 和 $G^{XB}$ ，利用这些公共值以及HDR报头中的CKY字段、SA和ID报体值，A和B可以分别计算哈希值 $HASH_A$ 和 $HASH_B$ ，通过签名得到 $SIG_A$ 和 $SIG_B$ 。
- 报文M5和M6的报体，即ID、CERT和SIG报体，都是被加密的密文。这两个报文中的CERT报体是可选项，可以选择传递证书报体。



## IKE交换1(续2)

- 采用数字签名真实性验证的自信模式交换过程如下：

M1:  $A \rightarrow B$ : HDR, SA, KE,  $N_A$ ,  $ID_A$

M2:  $B \rightarrow A$ : HDR, SA, KE,  $N_B$ ,  $ID_B$ , [CERT,]  $SIG_B$

M3:  $A \rightarrow B$ : HDR, [CERT,]  $SIG_A$

- 自信模式交换的报体与主模式完全相同，只是每个报文携带的报体组合不同。但是，报体的含义以及在真实性验证中的作用相同。



## IKE交换2：公钥加密真实性验证 的第一阶段协商

- 交换双方利用私钥解密对方加密的一次性数和标识，构造自己的哈希值发送给对方，证明自己的身份；重构对方的哈希值验证对方的身份。主模式的交换过程如下：

M1: A  $\rightarrow$  B: HDR, SA

M2: B  $\rightarrow$  A: HDR, SA



## IKE交换2(续1)

M3:  $A \rightarrow B$ : HDR, KE, [HASH(1),]  $PK_B\{\langle ID_{A1} \rangle\}$ ,  
 $PK_B\{\langle N_A \rangle\}$

M4:  $B \rightarrow A$ : HDR, KE,  $PK_A\{\langle ID_{B1} \rangle\}$ ,  $PK_A\{\langle N_B \rangle\}$

M5:  $A \rightarrow B$ : HDR\*,  $HASH_A$

M6:  $B \rightarrow A$ : HDR\*,  $HASH_B$

- 关键在于采用公钥加密传递密钥材料，通过返回的哈希值向对方验证已经获取了密钥材料。由此证明对方持有该公钥对应的私钥。

# IKE交换4：预共享密钥真实性验证的第一阶段协商

- A和B可以利用双方已经具有的共享密钥相互进行真实性验证。采用预共享密钥真实性验证的主模式交换过程如下：

M1: A  $\rightarrow$  B: HDR, SA

M2: B  $\rightarrow$  A: HDR, SA

M3: A  $\rightarrow$  B: HDR, KE,  $N_A$

M4: B  $\rightarrow$  A: HDR, KE,  $N_B$

M5: A  $\rightarrow$  B: HDR\*,  $ID_{A1}$ ,  $HASH_A$

M6: B  $\rightarrow$  A: HDR\*,  $ID_{B1}$ ,  $HASH_B$

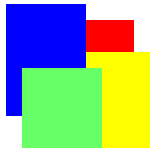


## IKE交换4(续1)

- IKE交换4的关键验证数据是双方已经具有的共享密钥:
- 共享密钥  $\text{SKEYID} = \text{PRF}(K_{A,B}, \langle N_A \rangle \parallel \langle N_B \rangle)$ , 这里  $K_{A,B}$  表示A和B预先共享的密钥。

$$\text{HASH}_A = \text{PRF}(\text{SKEYID}, G^{XA} \parallel G^{XB} \parallel \text{CKY}_A \parallel \text{CKY}_B \parallel \langle \text{SA}_A \rangle \parallel \langle \text{ID}_{A1} \rangle)$$

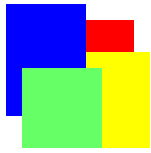
$$\text{HASH}_B = \text{PRF}(\text{SKEYID}, G^{XB} \parallel G^{XA} \parallel \text{CKY}_B \parallel \text{CKY}_A \parallel \langle \text{SA}_A \rangle \parallel \langle \text{ID}_{B1} \rangle)$$



# 基于IPsec的综合应用题

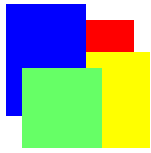
例题1：假设某个销售人员在外地试图通过公共互联网从公司网络服务器中下载销售资料 and 文件。请问：

- (1) 如果该销售人员不采用任何安全技术直接从公司服务器取数据会遇到哪几种安全威胁？
- (2) 该销售人员考虑采用安全技术，则他至少应该从哪几个方面考虑安全技术？



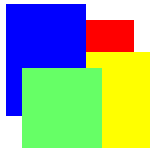
## 基于IPsec的综合应用题(续1)

- (3) 如果他试图保密地传递客户订单，他应该选用何种IPsec中的安全协议？
- (4) 如果他试图完整地传递客户促销计划，他最好选用哪种IPsec中的安全协议？
- (5) 如果他选择安全IP技术(IPsec)作为安全防范技术，他如何进行真实性验证？



## 基于IPsec的综合应用题(续2)

- (6) 如果公司服务器提供了访问控制机制，需要根据用户标识控制对公司服务器的访问权限，这时他应该选择哪种类型的安全关联建立协议？
- (7) 假定用户已经获得包括自己私钥和公司服务器公钥的证书，试具体描述这种用于真实性验证和安全关联建立的协议的主要步骤。



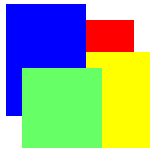
## 基于IPsec的综合应用题(续3)

(1) 如果该销售人员不采用任何安全技术直接从公司服务器取数据会遇到哪几种安全威胁？

答：如果不采取安全防范技术，他获取的销售资料可能被窃取，可能被更改，可以被假冒，可能无法访问到公司的服务器。

(2) 该销售人员考虑采用安全技术，则他至少应该从哪几个方面考虑安全技术？

答：他至少应该从真实性验证、访问控制和攻击检测三个方面考虑对安全技术的选择。



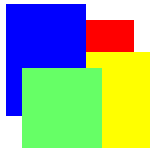
## 基于IPsec的综合应用题(续4)

(3) 如果他试图保密地传递客户订单，他应该选用何种IPsec中的安全协议？

答：为了保密地传递数据，应用IPsec技术时，必须采用ESP协议。

(4) 如果他试图完整地传递客户促销计划，他最好选用哪种IPsec中的安全协议？

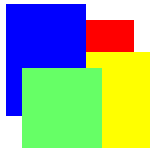
答：为了完整地传递数据，应用IPsec技术时，最好采用AH协议。



## 基于IPsec的综合应用题(续5)

(5) 如果他选择安全IP技术(IPsec)作为安全防范技术, 他如何进行真实性验证?

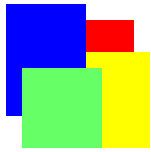
答: 他为了进行真实性验证, 必须首先通过其他安全途径(例如在离开公司之前, 直接到公司人力资源部)获得与他相关的用户标识和密钥(可以是公钥也可以是共享密钥)等个人身份数据, 然后, 在第一次网络连接过程中输入相关个人身份数据。



## 基于IPsec的综合应用题(续6)

(6) 如果公司服务器提供了访问控制机制，需要根据用户标识控制对公司服务器的访问权限，这时他应该选择哪种类型的安全关联建立协议？

答：由于用户标识是系统进行访问控制的依据，所以，在应用IPsec技术创建安全关联时，必须采用具有用户标识保护功能的安全关联创建协议。



## 基于IPsec的综合应用题(续7)

(7) 假定用户已经获得包括自己私钥和公司服务器公钥的证书, 试具体描述这种用于真实性验证和安全关联建立的协议的主要步骤。

答: 运用IKE协议, 选用主模式下基于公钥真实性验证的安全关联创建协议如下:

M1: A  $\rightarrow$  B: HDR, SA

M2: B  $\rightarrow$  A: HDR, SA

M3: A  $\rightarrow$  B: HDR, KE, [HASH(1),]  $PK_B\{<ID_{A1}>\}$ ,  $PK_B\{<N_A>\}$

M4: B  $\rightarrow$  A: HDR, KE,  $PK_A\{<ID_{B1}>\}$ ,  $PK_A\{<N_B>\}$

M5: A  $\rightarrow$  B: HDR\*,  $HASH_A$

M6: B  $\rightarrow$  A: HDR\*,  $HASH_B$



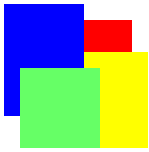
# 传送层安全加固

- 传送层安全加固是指在端到端的传送层进行安全加固的技术。
- 传送层安全加固是直接面向网络应用的安全加固技术。它是网络安全技术中较为成功的一类网络安全加固技术。
- 现在广泛使用的传送层安全协议主要是指安全套接层(SSL)协议和在SSL协议基础上标准化的传送层安全(TLS)协议，SSL协议最初是为安全访问万维网(WWW)应用而设计的。



# SSL协议概述

- 安全套接层协议是一种提供在互联网环境下**秘密通信**的安全协议。
- 该协议允许客户端和服务端应用在防范窃听、干扰和假冒报文的方式下进行通信。
- 现在通常使用的SSL协议的版本是**SSL 3.0**。
- SSL协议的一个**主要优点**是**独立于应用协议**，一个应用协议可以透明地加载到SSL协议之上。

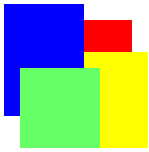


# SSL协议概述(续1)

- SSL协议包括两个层次：
  - 记录协议，直接基于某个可靠传送协议之上用于封装SSL上层协议；
  - 握手协议、告警协议、加密规范更改协议、以及应用数据协议。

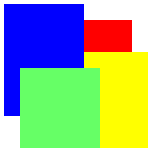
握手协议	告警协议	加密规范更改协议	应用数据协议
记录协议			

SSL协议分层结构



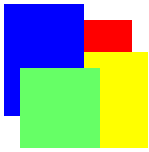
## SSL协议概述(续2)

- SSL协议提供的连接安全具有以下3种特征：
  - 连接是秘密的。SSL协议经过初始握手协商了通信双方的共享密钥之后，就可以利用SSL记录协议对所有应用数据进行加密传递。SSL协议采用对称密钥加密算法。
  - 对等方的身份是可以验证的。SSL协议利用 公钥加密算法进行应用层通信双方的真实性验证。
  - 连接是可靠的。在SSL连接上传递的报文包括了报文验证码，可以用于报文完整性检查。



# SSL协议之间关系

- SSL协议完成握手协议交互，建立SSL会话之后，就可以传递应用层协议的报文。
- SSL协议发送和接收应用协议报文的过程，实际上就是SSL记录协议处理应用协议报文的过程。
- SSL记录协议对SSL上层协议，握手协议、告警协议和加密规范更改协议，也采用以上处理流程。



# SSL记录协议概述

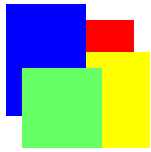
- SSL记录协议利用SSL会话和连接中协商的加密算法、真实性验证算法以及相应的密钥，对上层协议报文进行加密。SSL会话和连接必须通过SSL握手协议创建。
- SSL握手协议可以作为SSL协议中关键的信令协议，SSL记录协议可以作为SSL协议中关键的数据传递协议。

# SSL记录协议的处理过程(续2)

- SSL记录协议处理应用报文的过程:



SSL协议发送和接收报文的处理过程



# SSL握手协议概述

- SSL握手协议的交互过程可以简单概括为：
  - SSL客户端和服务端协商一个协议版本，
  - 选择加密算法，
  - 彼此验证身份(可选项)，
  - 并且利用公钥加密技术生成共享保密字。



# SSL的会话

- SSL会话是两个应用实体相互进行真实性验证之后建立的一种关联，这种关联类似于互联网安全关联与密钥管理协议(ISAKMP)中定义的ISAKMP安全关联。
- SSL会话协商了以下SSL会话属性：
  - 会话标识符，由服务器选择的一个任意字节序列，用于标识一个活跃的、可继续使用的会话。



# SSL的会话(续1)

- SSL会话协商了以下SSL会话属性：
  - 对等方证书，存放应用协议交互对等方公钥的X.509v3证书，该属性可能为空。
  - 压缩方法，SSL记录协议压缩数据使用的压缩算法。
  - 加密及真实性验证规范。
  - 主保密字，在客户端和服务端之间共享的48字节的保密字。
  - 可继续标志，指示该会话是否可以用于发起一条新连接的标志。



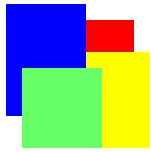
# SSL的连接

- SSL连接是在SSL会话之上，协商应用实体交互双方采用的真实性验证保密字、加密密钥、加密算法初始向量的一条安全连接。
- SSL连接类似于ISAKMP协议在第二阶段建立的安全协议关联。
- 一个SSL会话可以包括多条SSL连接，而服务器和客户机之间可以同时建立多个SSL会话，满足不同应用的安全通信需求。



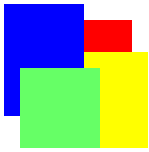
# SSL连接的属性

- SSL连接具有以下属性：
  - 服务器和客户机随机数，服务器和客户机为每个连接选择的一个字节序列，相当于服务器和客户机的一次性数。
  - 服务器写MAC保密字，服务器生成MAC时采用的保密字。
  - 客户机写MAC保密字，客户机生成MAC时采用的保密字。



## SSL连接的属性(续)

- SSL连接具有以下属性：
  - 服务器写密钥，服务器对成批数据加密的密钥；
  - 客户机写密钥，客户机对成批数据加密的密钥；
  - 初始向量，在CBC模式下使用块加密时，需要为每个密钥维护一个初始向量。
  - 顺序编号，对于每个连接，服务器和客户机都维护两个不同的顺序编号，用于发送和接收报文。



# 握手协议的交互过程

M1: A  $\rightarrow$  B: ClientHello

M2: B  $\rightarrow$  A: ServerHello[, Certificate] [,  
ServerKeyExchange]

[, CertificateRequest], ServerHelloDone

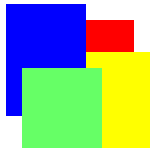
M3: A  $\rightarrow$  B: [Certificate,] ClientKeyExchange[,  
CertificateVerify],

ChangeCipherSpec, Finished

M4: B  $\rightarrow$  A: ChangeCipherSpec, Finished

M5: A  $\rightarrow$  B: ApplicationData

M6: B  $\rightarrow$  A: ApplicationData



# 握手协议的简化交互过程

- SSL客户端也可以利用已经建好的SSL会话创建新的SSL连接，简化握手协议：

M1: A → B: ClientHello

M2: B → A: ServerHello, ChangeCipherSpec, Finished

M3: A → B: ChangeCipherSpec, Finished

M4: A → B: ApplicationData

M5: B → A: ApplicationData



# 网络应用安全技术

## 网络应用安全解决方案

- 基于IPsec的安全解决方案
- 基于SSL的安全解决方案

## 电子邮件安全技术

- PGP安全技术
- S/MIME安全技术

## 万维网(WWW)安全技术

- 万维网攻击分析，SQL注入攻击、XSS攻击
- 万维网的安全防范技术

## 区块链及其应用



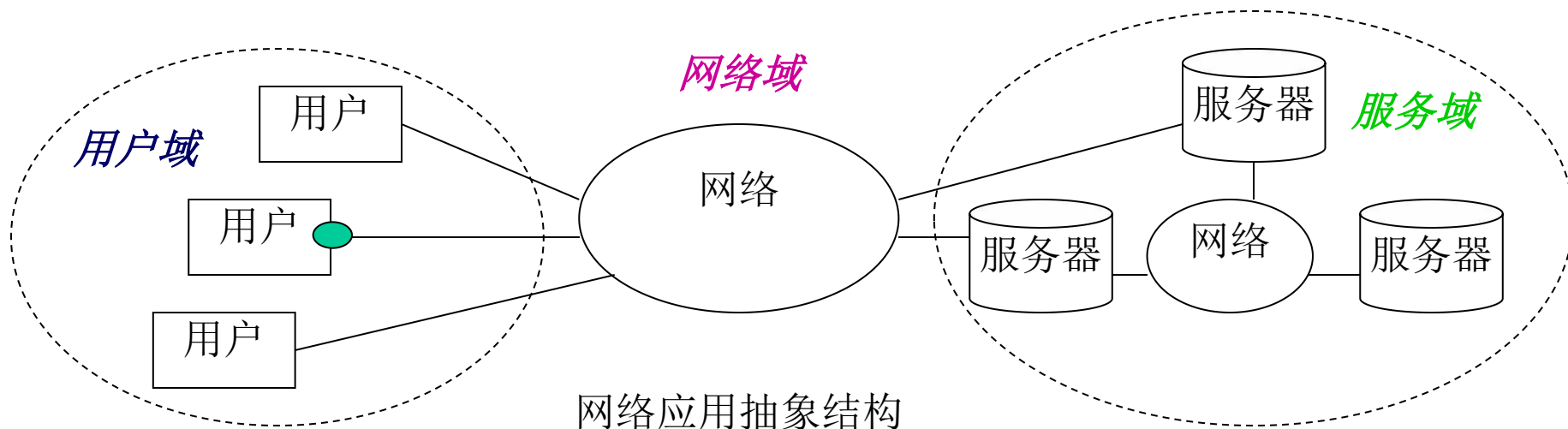
# 网络应用安全概述

- 网络安全的目标(内涵)包括：保密性、完整性和可用性。因此，网络应用安全技术需解决以下两方面问题：
  - 1) 一方面，需要解决网络应用相关数据的保密性和完整性问题，这里方面的责任人包括：应用系统管理员、用户。
  - 2) 另一方面，需要解决网络应用系统本身的可用性问题，这方面的责任人包括：网络系统和应用系统管理员。



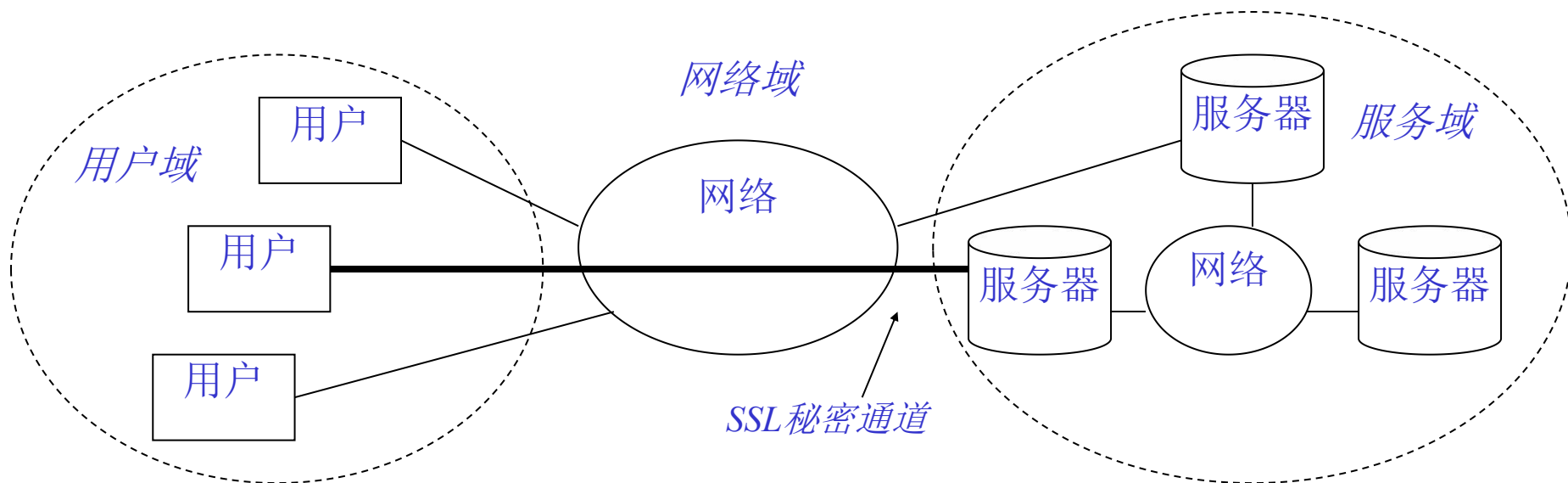
# 网络应用的抽象结构

- 网络应用结构包括了用户域、网络域和服务域。



# 基于SSL的网络应用安全方案

- 网络应用安全中的保密性和完整性方案：  
基于SSL的安全方案，适用于用户域和服务域都不安全的应用环境。

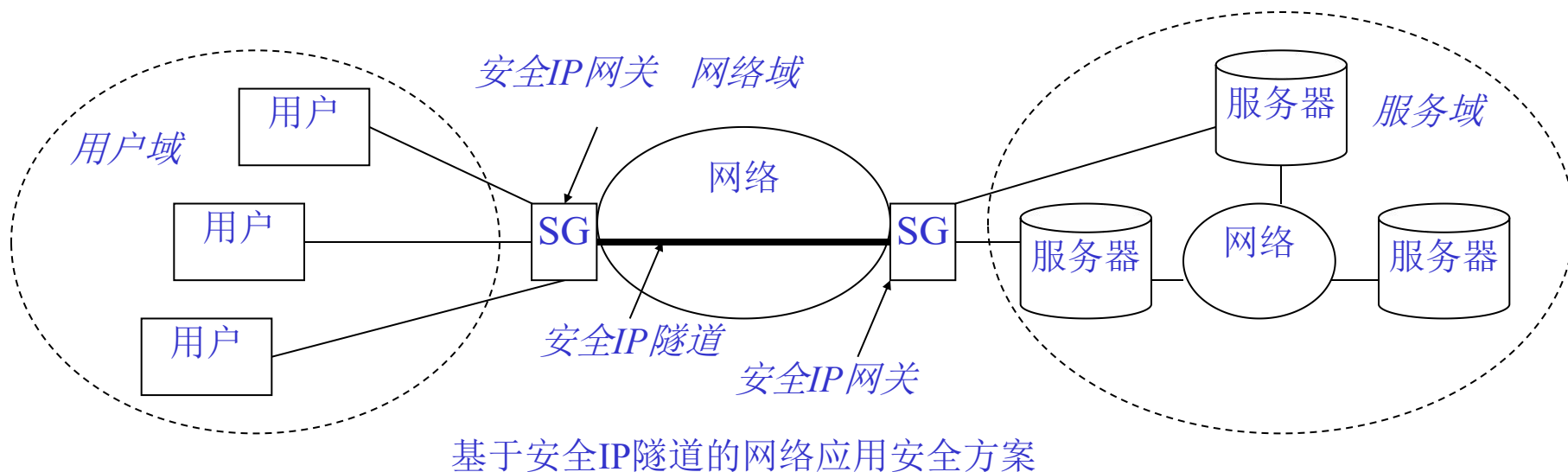


基于SSL的网络应用安全方案



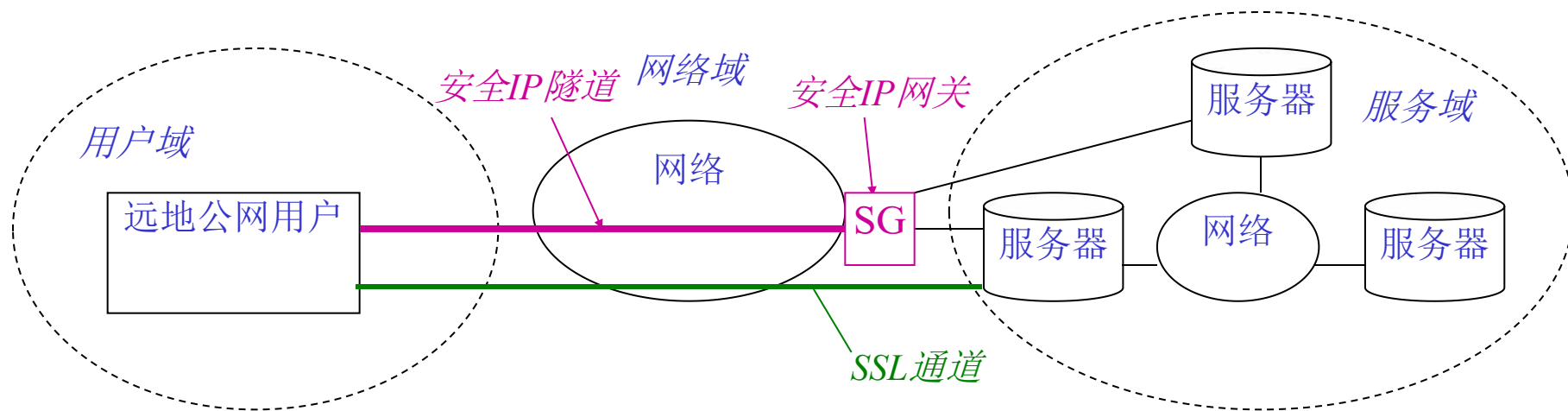
# 基于安全IP隧道的方案

- 网络应用安全中的保密性和完整性方案：  
基于安全IP隧道的方案，适用于用户域和服务域都安全的应用环境。



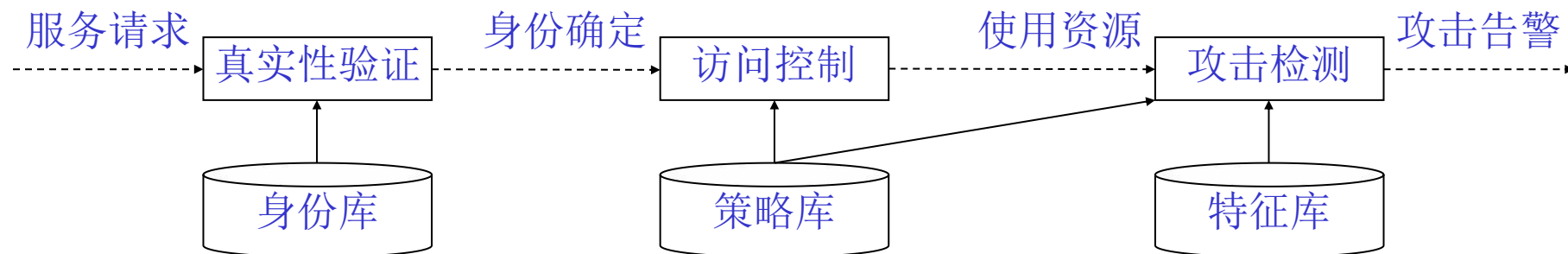
# 远地网络用户的安全解决方案

- 单个远地用户：可以采用基于SSL的安全方案，也可以采用安全IP隧道方案。



# 网络应用的可用性解决方案(1)

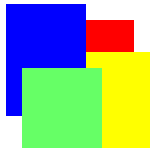
- 网络应用安全中的可用性需要综合运用真实性验证、访问控制和攻击检测技术进行保护(防攻击技术)。
- 基于真实性验证、访问控制和攻击检测的网络应用的可用性保护模型(防攻击模型)。





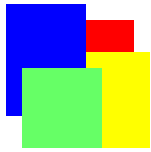
## 网络应用的可用性解决方案(2)

- 真实性验证的结果可能是合法的网络实体，也可能是匿名的网络实体，或者非法的网络实体对网络进行访问。
- 按照真实性验证的结果进行访问控制：
  - 如果访问控制策略表中没有匿名用户或者匿名实体项，则说明该安全系统不对匿名用户开放；
  - 如果访问控制策略表中没有指定合法用户的控制项，则说明该安全系统不向该合法用户开放。
- 用户在访问网络和使用网络资源过程中，攻击检测将进行实时检测和事后检测。攻击检测是按照访问控制策略库和网络攻击特征库，采用智能算法进行的。



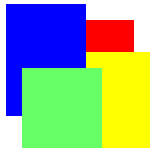
# 电子邮件安全应用技术

- 电子邮件是因特网上使用最为普遍的网络应用之一。
- 从网络安全角度看，目前电子邮件是造成网络安全威胁的一个主要渠道。研究和开发电子邮件安全技术不仅具有很大的实用价值，而且具有较大的研究意义。
- 目前电子邮件安全技术主要解决完整地、保密地传递电子邮件。
- 目前常用的安全传递电子邮件的技术包括完美隐私(英文缩写PGP)技术和安全多用途因特网邮件扩展(S/MIME)技术。



# 完美隐私(PGP)技术

- 完美隐私(Pretty Good Privacy, PGP)技术是一个典型的面向网络应用的安全技术，它是报文身份验证技术和报文加密技术在电子邮件和文件存储方面具体的应用。
- PGP技术是美国麻省理工学院(MIT)软件工程师Phil Zimmermann个人发明并且推广应用的网络安全应用技术。



# 完美隐私(PGP)功能与应用

- PGP技术目前已经在所有常用的计算机操作系统上实现，其中既有PGP实现的自由软件，又有PGP实现的商业软件。
- PGP提供了5种与网络应用安全相关的服务：报文真实性验证、报文加密、报文压缩、安全电子邮件、以及报文分段服务。
- PGP服务也可以在网络环境下的非电子邮件应用中提供。



# PGP报文真实性验证的发送方

- PGP采用SHA-1报文摘要算法与RSA公钥加密算法结合，实现对报文的真实性验证。具体的过程如下：

$$M1: A \rightarrow B: M \parallel VK_A\{H(M)\}$$

- 说明：发送方A对报文M执行报文摘要算法SHA-1，得到H(M)，然后利用A的私钥 $VK_A$ 对H(M)进行加密(数字签名)，得到 $VK_A\{H(M)\}$ ，将其附加在报文M后发送给接收方B。
- 目的：杜绝假冒的电子邮件

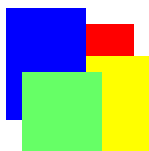


# PGP报文真实性验证的接收方

- PGP采用SHA-1报文摘要算法与RSA公钥加密算法结合，实现对报文的真实性验证，过程如下：

$$M1: A \rightarrow B: M \parallel VK_A\{H(M)\}$$

- 接收方B收到报文M之后，利用发送方A的公钥解密报文验证码，获得发送过来的报文摘要H(M)。B同时利用SHA-1算法重新计算收到的报文M的摘要H'(M)，如果H'(M) = H(M)，则B可以验证报文M是真实的。
- 数字签名的安全性是由公钥的真实性保证，而这种真实性的保证可以通过一个实名群的公开发布实现。



# PGP的报文加密方法

- PGP的报文加密方法不需要协商密钥
  - PGP的报文加密方法不采用安全IP技术和SSL安全技术中的密钥协商方法，而是利用公钥加密方法传递传统加密算法使用的对称密钥(称为会话密钥)，利用传统加密算法加密报文。
- 每个会话密钥只用于加密一个报文。所以，PGP是采用“一次一密钥”的方法加密报文，这是最为安全的报文加密方法。



# PGP的报文加密过程

- PGP对报文的加密过程如下：

(1) 对于某个等待发送的报文M，发送方A随机产生一个对称密钥 $K_{A,B}$ ，并对M加密：

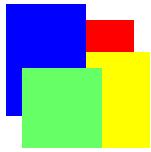
$$M1: A \rightarrow B: K_{A,B}\{M\} \parallel PK_B\{K_{A,B}\}$$

(2) 接收方B收到报文M1之后，首先用自己的私钥解密 $PK_B\{K_{A,B}\}$ ，得到本次加密报文的对称密钥 $K_{A,B}$ ，然后，再利用 $K_{A,B}$ 解密 $K_{A,B}\{M\}$ ，得到报文M。



# PGP的报文压缩

- PGP采用ZIP算法进行报文压缩。
- PGP对报文的处理顺序：
  - 报文签名→报文压缩→报文加密
- 选择操作处理顺序的考虑：
  - 如果加密或压缩后签名，无法直接验证明文。
  - 压缩后加密提高加密效率及数据保密性。
  - 电子邮件加密/解密是客户端到客户端应用，客户端的身份验证无需防范DOS攻击！
- 安全IP技术的操作处理顺序：
  - 报文加密→报文签名（防DOS攻击）
  - 安全IP技术属于网络层安全防范技术，必须防范DOS攻击



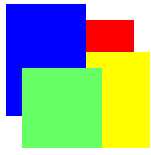
# PGP的安全电子邮件服务

- 将报文加密算法和报文真实性验证算法应用于电子邮件存在的障碍：
  - 传统电子邮件系统只能传送7bits的ASCII码正文字符。
  - 加密算法和真实性验证算法都是生成以8bits为单位的任意数据流。
- 采用radix-64转换算法，将3个8bits数据转换成4个7bitsASCII码字符。



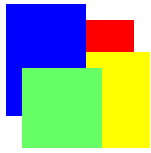
# PGP的报文分段服务

- 传统因特网上的电子邮件系统会限定电子邮件的长度，为解决这个问题，PGP提供了报文分段的服务。
- 发送方PGP将较大的电子邮件分为较小的电子邮件，然后将分段后的电子邮件逐一发送出去。接收方PGP将分段电子邮件进行合段后再做其他处理。



# PGP对电子邮件报文的处理流程

- 电子邮件发送方处理流程：
  - 发送用户数据 → 签名 → 报文压缩 → 报文加密 → 密文转换为ASCII码字符流 → 分段大报文 → 发送电子邮件
- 电子邮件接收方处理流程：
  - 接受电子邮件 → 报文合段 → ASCII字符流转换为密文 → 报文解密 → 报文解压缩 → 报文真实性验证 → 用户提交数据



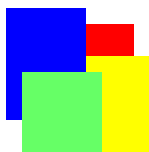
# 万维网(WWW)安全应用技术

- 万维网面临的安全威胁
- 万维网安全防范技术
- 万维网攻击检测技术
- SQL（结构化查询语言）注入攻击
- XSS（跨网站脚本编写）攻击



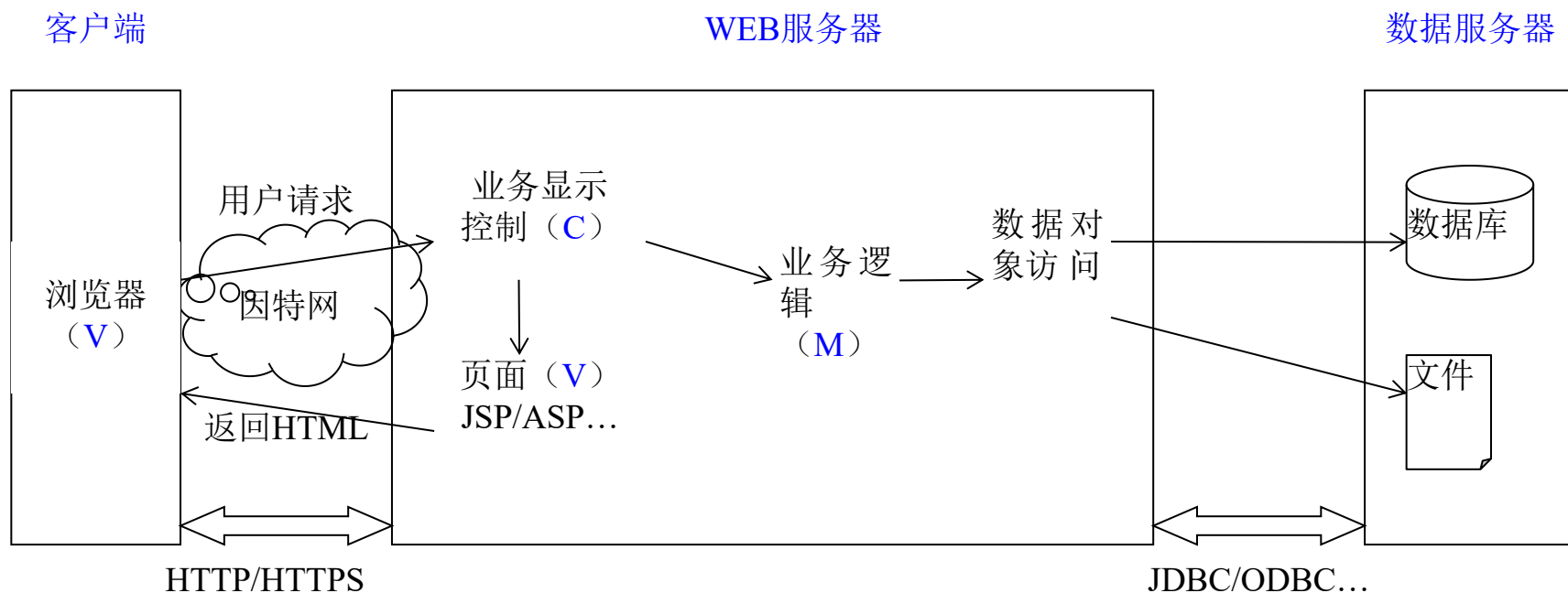
# 万维网(WWW)安全技术

- 万维网应用由三个部分构成：
  - 提供万维网服务的万维网服务器
  - 使用万维网服务的万维网浏览器
  - 传递浏览器和服务端之间服务请求和响应报文的网络。
- 这三个部分都面临安全的威胁。
- 为何特别讨论万维网的安全技术？



# 万维网(WWW)应用系统框架

- 基于MVC (Model-View-Controller, 模型-视图-控制器)设计模式的万维网应用系统框架





# 万维网服务器安全漏洞

- 万维网服务器安全漏洞主要源于两个方面：
  - 其一，万维网服务器软件错误产生的安全漏洞
  - 其二，万维网服务器配置不当产生的安全漏洞
- 这些安全漏洞都会允许远地网络用户非法进入万维网服务器，窃取非公开的数据和文件，执行非授权的修改系统的命令，发起“拒绝服务”攻击，使得万维网服务器泄露敏感数据或处于“不可用”状态。



# 针对万维网服务器的攻击举例

- (1) 身份真实性验证类攻击：以攻击万维网服务器的真实性验证系统为目标。如：
- 穷举口令类攻击
  - 破译弱强度口令类攻击
  - 口令恢复验证类（保密问题或口令暗示）攻击



# 针对万维网服务器的攻击举例

(2) 授权访问类攻击：以攻击万维网服务器的访问控制系统为目标。例如：HTTP会话劫持攻击。

- 用户第一次通过身份真实性验证与web服务器建立连接后，服务器会产生一个会话标识S-ID保存在cookie中，以此作为后续通信的验证信息来区别各类用户的访问权限。
- 攻击者可以通过ARP(地址解答协议)等欺骗手段，将双方交互的报文发送至攻击者，截获S-ID，冒充真实的用户获取非法的授权访问能力，劫持已建立的会话。



# 针对万维网服务器的攻击举例

(3) 注入代码类攻击：以攻击万维网服务接口为目标。

– SQL注入、SSI注入、Xpath注入、LDAP注入等， 如：

`select * from products where productID = '+用户输入ID号+'`

如果用户输入16，则该语句在执行时变为：

`Select * from products where product_id = ' 16 '`

数据库会将ID为16的商品信息返回给用户。

如果攻击者输入： `' or '1'='1` 结果？ 获取所有商品信息

`select * from products where ' or '1'='1'`



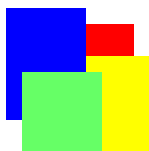
# 针对万维网服务器的攻击防范措施

- 针对服务器验证类攻击，可以采用强用户密码、限制猜测次数、多类型真实性验证机制等防范措施。
- 针对授权访问控制类攻击，可以采用基于角色的访问控制、cookie加密传输等防范措施。
- 针对注入代码类攻击，可以采用输入数据强验证等防范措施。
  - 软件编程人员一定要养成对于所有输入数据进行检查的编程习惯，这样才能编制出“可信”软件



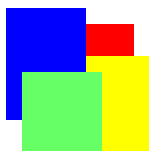
# SQL注入攻击定义

- SQL（结构化查询语言）注入攻击，是目前最为有效的入侵网站的数据库、窃取用户敏感信息的攻击。
- SQL利用万维网应用软件中没有严格过滤或验证输入数据而产生的漏洞，通过万维网应用提供的用户数据输入的功能，注入恶意软件，窃取网站上的用户登录账户和密码，进一步假冒网站合法用户登录网站，窃取网站的其他机密数据。



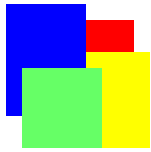
# SQL注入攻击分类

- SQL注入攻击可以分成4种类型：
- **SQL操纵类攻击**，利用不同的SQL操作，例如SQL的UNION(关系的并)操作，修改SQL语句的攻击过程。
  - 例如：通过更改WHERE语句的相关判定条件，使得WHERE语句总是为真，或者总是为假，实现SQL注入。
- **语句插入类攻击**，插入新的SQL语句的攻击过程。这类攻击方式仅当单个数据库的查询请求支持多个SQL语句时才能够发起攻击。
  - 例如在易攻击的SQL语句后面附加一个SQL服务器的EXECUTE（执行）命令。



# SQL注入攻击分类(续)

- 功能调用类攻击，在某个易攻击的SQL语句中插入不同数据库功能调用的攻击过程。这些插入的功能调用可以发起操作系统的调用或者操纵数据库中的数据。
- 缓存溢出类攻击，针对网络应用程序的漏洞常用的一类攻击。在SQL注入攻击中，这类攻击借助于SQL的功能调用注入，利用系统中的漏洞，导致缓存溢出，进而发起攻击。
  - 这类攻击仅仅对存在漏洞、并且没有及时打补丁的服务器才能发起有效的攻击。



# 重复式查询的SQL攻击

- 重复式查询属于SQL操纵类攻击，攻击者可以基于SQL的条件语句，在查询语句中注入恶意代码。例如一个登录的查询如下：

```
SELECT * FROM User_Info WHERE UserName = 'Bob'  
and Password='123456'.
```

- 在这条登录查询语句中，可以注入OR 1=1'，结果的登录查询语句如下：

```
SELECT * FROM User_Info WHERE UserName = 'OR  
1=1';-- and Password='123456'.
```



# 逻辑错误查询的SQL攻击

- 逻辑错误查询也属于SQL操纵类攻击，在这类攻击中，攻击者利用数据库服务器返回的出错消息，获取有关数据库内部的敏感信息。
- 例如以下查询语句可获取有关数据库表结构的敏感信息，例如数据库的表名、表的属性名（表的列名）等信息。

```
SELECT * FROM User_info WHERE UserName = ' HAVING  
1=1';-- and Password='123456'.
```

- 返回出错消息“列 ‘User\_info.UserID’ 在选择列表中是无效的，因为它既没有包含在一个汇聚函数中，也没有包含在GROUP BY语句中。”，由此可获得该数据库的表名：User\_info，以及该表对应的属性名：User\_info.UserID



# 合并查询的SQL攻击

- 合并查询属于语句插入类和SQL操纵类组合攻击。它在安全的查询上附加恶意代码，用于获取其他的表信息。例如可以获取表属性的数据类型信息。例如数据库服务器执行以下的查询语句：

```
SELECT * FROM User_info WHERE UserName = ' UNION  
SELECT SUM(Uername) from User_info-- BY UserID  
HAVING 1=1';-- and Password='123456' .
```

- 返回出错消息：“对于SUM操作符，操作数的数据类型 *VARCHAR* 无效”，该出错消息就泄露了Uername的数据类型是*VARCHAR*的信息。

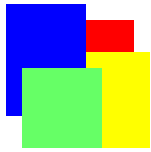


# 背驮式查询的SQL攻击

- 背驮式查询攻击属于语句插入类攻击。攻击者在传统的查询语句中注入恶意代码，并且也执行数据库操纵类操作，例如INSERT（插入）、UPDATE（更新）、DELETE（删除）语句，操纵某个数据库中的记录。例如以下查询语句：

```
SELECT * FROM User_info WHERE UserName =  
';INSERT INTO User_info VALUES('Bob','123')-'
```

- 这样就可将某个非法的记录插入到数据库的User\_info表中。



# 真实性验证防范SQL攻击

- **基本思路**：SQL注入攻击中的SQL语句都是假冒的SQL语句，通过在SQL语句中关键操作符和用户相关信息进行加密，防范假冒的SQL语句，以此防范SQL注入攻击。
- **具体方案**：每个用户赋予一个传统加密算法的密钥，服务器则赋予用于公钥加密算法的公钥和私钥。对于SQL的登录查询，采用两级加密：
  - 采用传统加密算法和用户的密钥，加密用户名和登录密码；
  - 采用公钥加密算法和服务器的公钥，加密查询模式。



# 真实性验证防范SQL攻击(续)

- 执行过程包括三个阶段：
  - 注册阶段，用户注册服务器，获得赋予的用户密钥；
  - 登录阶段，采用加密算法加密SQL登录查询语句；
  - 验证阶段，服务器接收到用户的SQL登录查询语句之后，通过解密算法进行SQL登录查询语句的真实性验证。
- 只有通过真实性验证的SQL登录查询语句，才提交SQL服务器执行，以此防范SQL注入攻击。经过测试，这种真实性验证方案十分有效，加密和解密过程可以在不到一秒钟时间内完成。
- 方案的不足：无法防范基于链接的SQL注入攻击；难以维护客户端的传统加密算法的密钥、以及服务器端的公钥加密算法的密钥；在注册阶段缺少安全保护机制。



# SQL查询属性值移除方案

- 基本思路：在用户提交的SQL查询语句的属性值中，移除SQL查询语句的方案，消除攻击者通过SQL查询语句属性值的代码注入，防范某种类型的SQL注入攻击。
- 采用“异常检测”的方法，通过统计分析，获取正常用户的SQL查询的特征样本(查询简本)。通过正常用户的SQL查询简本与攻击者动态产生的SQL查询进行比对，检测并识别SQL注入攻击。
- 方案不足：采用统计方法获取正常用户使用SQL查询的行为特征，行为特征描述不完整、不准确，可能产生误判，影响正常的SQL查询操作。



# 跨网站脚本编写(XSS)的攻击

- 跨网站脚本编写（XSS）是另一种常见的万维网应用攻击技术，通过用户浏览过的网页，将攻击者的恶意代码传回到该用户的浏览器或客户端。该脚本将在该用户浏览器的安全区域运行，可以读取、更改、传送该浏览器可以访问的任何关键数据。
- XSS攻击可以分成两类：持续式XSS和非持续式XSS。
  - 当恶意代码成功地植入某个万维网应用（网页邮箱、论坛等）中，则就会出现存储式/持续式XSS攻击。持续式XSS攻击不需要引诱用户点击任何网络的链接。
  - 当服务器没有很好地检测和清除万维网页面中隐藏的恶意网络链接时，就可能发生反应式或非持续式XSS攻击。恶意链接包含的恶意代码，在用户点击后就可以下载到用户的浏览器并且运行，使得攻击者可以获取用户敏感的数据

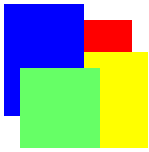
# XSS攻击举例

## (1) 持续式XSS攻击

例如：跨站点脚本攻击



图7.7 跨站点脚本攻击过程



# XSS攻击举例(续)

## (2) 非持续式XSS攻击

例如：某页面所在地址为

```
<frame src = "http://truth.example/file.html">
```

该地址也可以通过URL参数形式定义为

```
http://truth.example/page?frame_src=http://truth.example/file.html
```

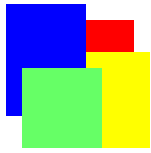
攻击者可以将frame\_src参数替换为

```
<frame_src=http://attacker.example/spoof.html>
```

当用户点击链接后

浏览器地址栏中显示http://turth.example

而实际链接却指向了http://attacker.example。



# 保障万维网服务器的可用性

- 在公共万维网服务器前端设置一个万维网服务器防火墙，用于过滤异常的万维网服务请求，保护万维网服务器不受“拒绝服务”这类攻击。
- 万维网防火墙是专门针对万维网服务器设计的一个访问控制系统，它本身携带安全策略库和审核数据库，用于控制进出万维网服务器的报文。
- 比一般的网络防火墙针对性强，可以制定较为详细的、有针对性的安全控制策略，并且可以通过及时调整安全控制策略，防范潜在的网络攻击。



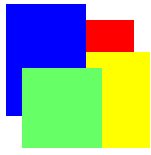
# 保障万维网浏览器的可用性

- 防范措施：用于过滤进入浏览器的数据，包括Java脚本等移动代码，通过特征分析剔除可能的恶意代码。例如：
  - (1) 管住自己，不要随意访问不熟悉的网站；
  - (2) 禁用ActiveX插件、控件和Java脚本
  - (3) 安装防病毒软件
  - (4) 注册表加锁
  - (5) 禁用远程注册表操作服务
  - (6) 及时升级万维网浏览器或选用安全浏览器



# 区块链及其应用

- 比特币与区块链
  - 比特币是全球首先成功使用区块链构建的去中心化数字货币系统
- 区块链的“块”和“链”
  - 区块链的“块”是具有真实性验证能力的块
  - 区块链的“链”是具有真实性验证能力的链
- 区块链的应用
  - 比特币是区块链在金融领域的成功应用
  - 区块链可以在互联网其他领域得到应用
  - 区块链应用目标：去中心化的信任管理



# 比特币：去中心化数字货币系统

- 比特币是一种基于互联网之上的覆盖网络、采用分布式控制方法构建的自治的、网络数字货币系统。
- 每个比特币系统的结点采用对等网络(P2P网络)方式接入比特币的网络，构成了一个对等的比特币的应用网络。
- 每个比特币系统的用户通过设立自己的钱包，将自己的公钥作为钱包地址，用于接收比特币；采用数字签名，花费自己钱包的比特币。
  - 用户在比特币系统结点设立钱包时获取密钥对，无需PKI



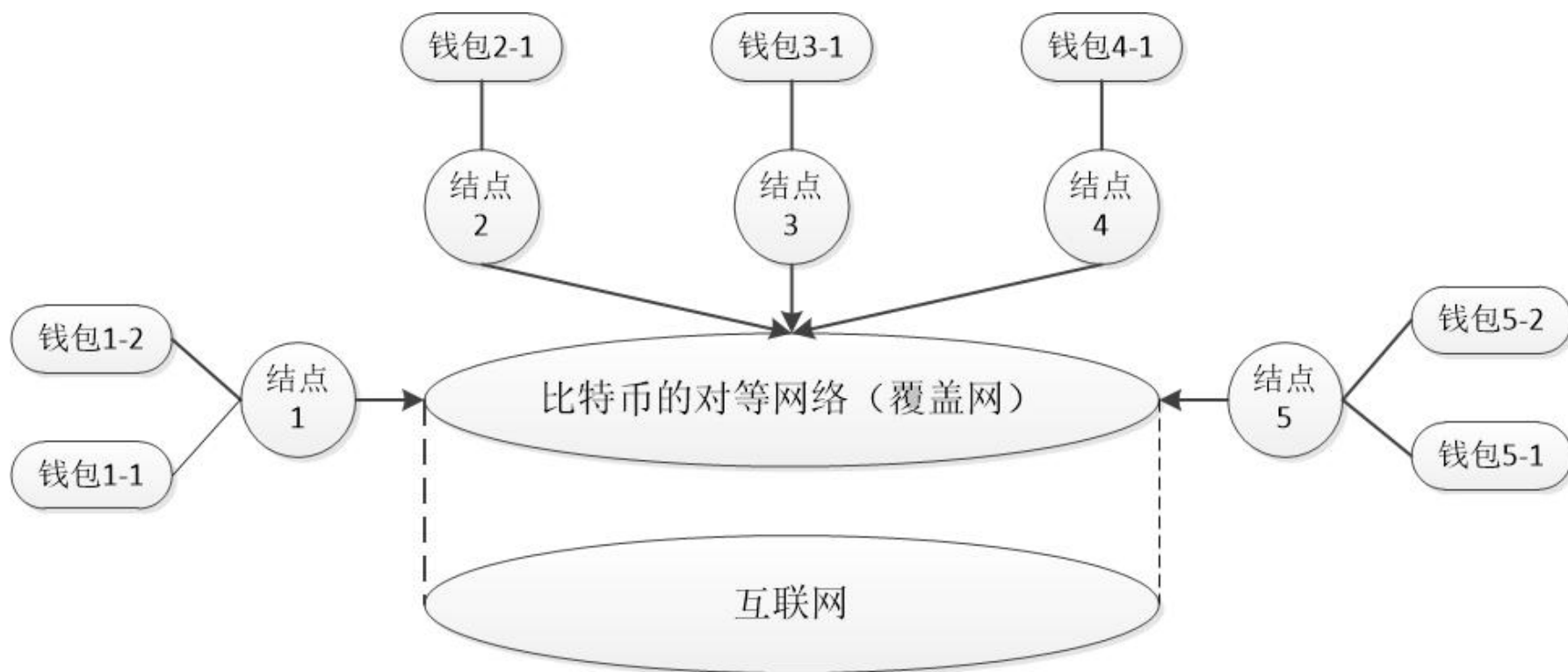
# 比特币与区块链

- 比特币系统巧妙地综合了四十多年来在密码学、数据真实性验证、以及对等网络等方面的已有研究成果，采用高度复杂的、原创的、实际可行的方法解决了去中心化的数字货币的女巫攻击和重复花费的难题。
  - 重复花费：将一个数字货币花费两次或多次
  - 女巫攻击：单个实体假冒多个实体投票，形成假冒多数
- 比特币系统采用区块链解决以上两个难题：
  - 采用对等网络通告所有交易，通过区块链记录有效交易
  - 当区块链出现分叉时，选择最长链；利用工作证明方式限制每个实体投票(验证)数目，防范假冒的有效交易验证。



# 比特币系统的示意图

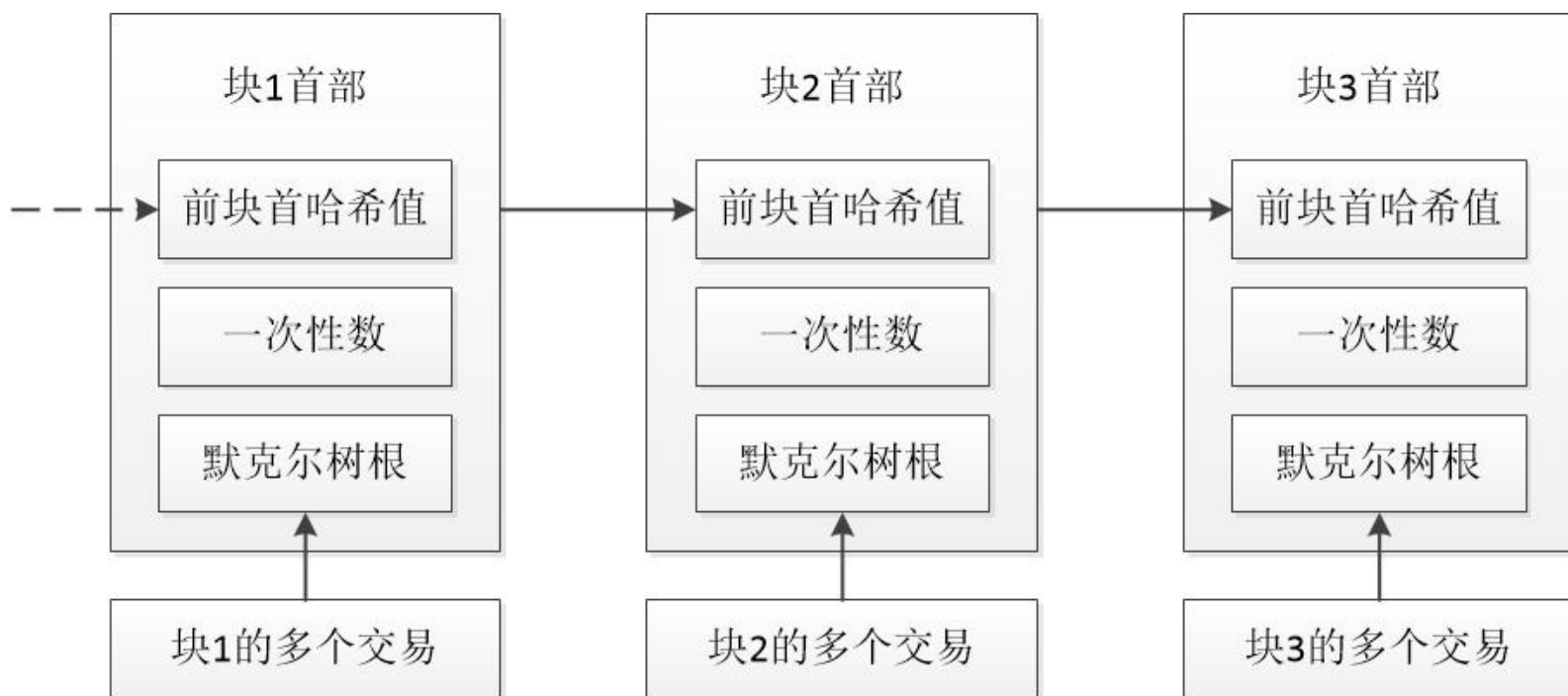
- 结点就是一台连接到比特币系统的计算机
- 每个结点可以有多个钱包





# 区块链是比特币的公共账本

- 区块链记录比特币的所有的交易账单。
- 比特币系统的每个结点保存和维护有效的所有区块链。
- 每个块记录了一次或多个交易。

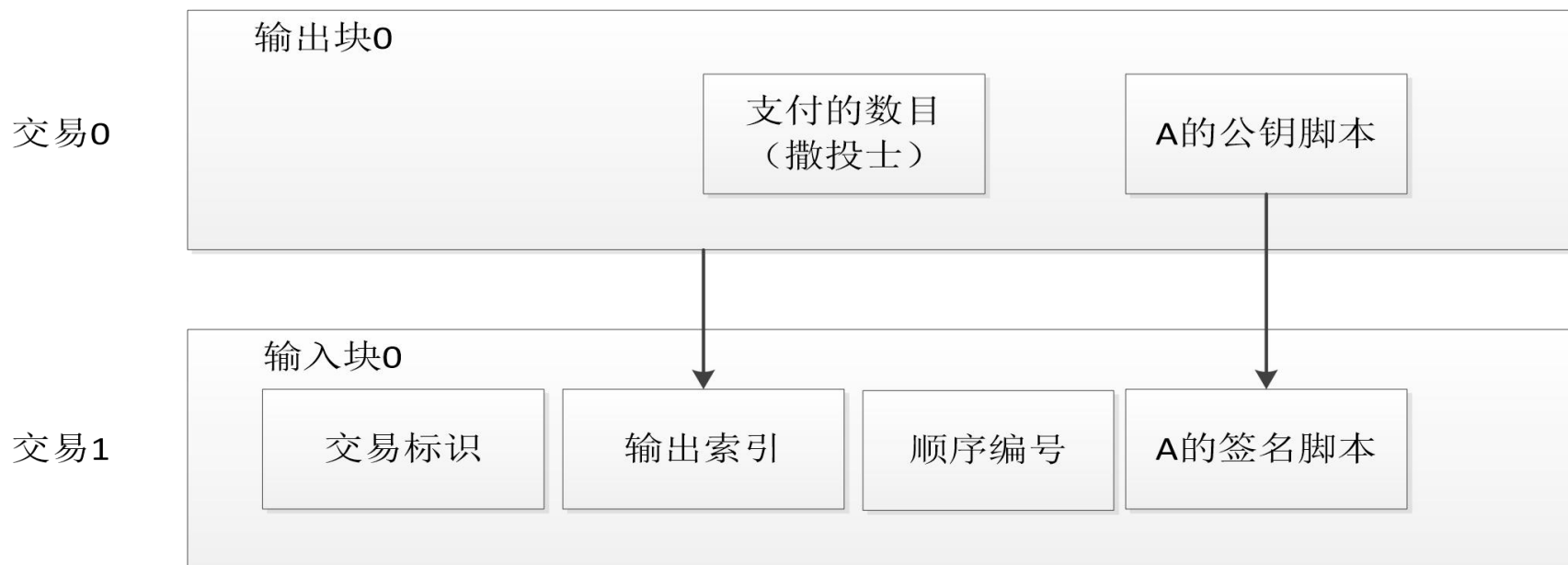




# 输入块和输出块——交易

- 输入块验证拟花费的比特币钱包所有者
- 输出块指定拟支付的比特币数目和对象

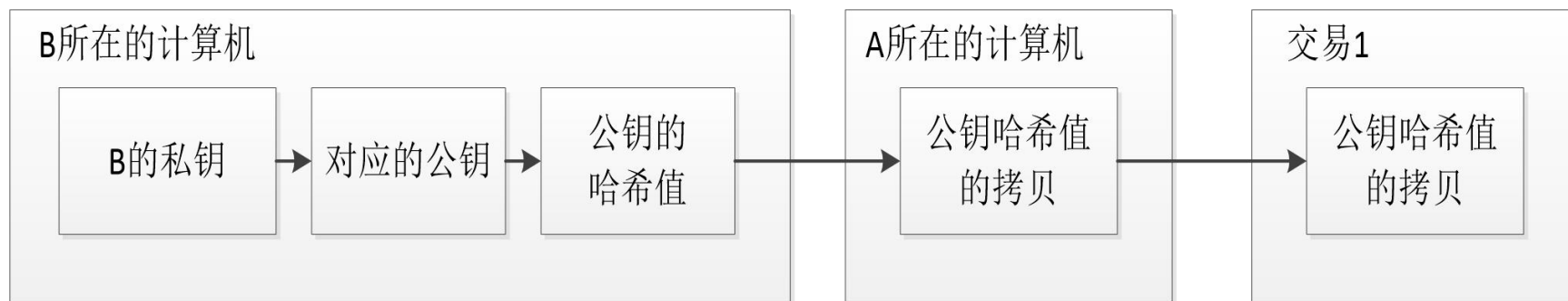
输出块举例：支付给A的公钥脚本





# 比特币的交易类型

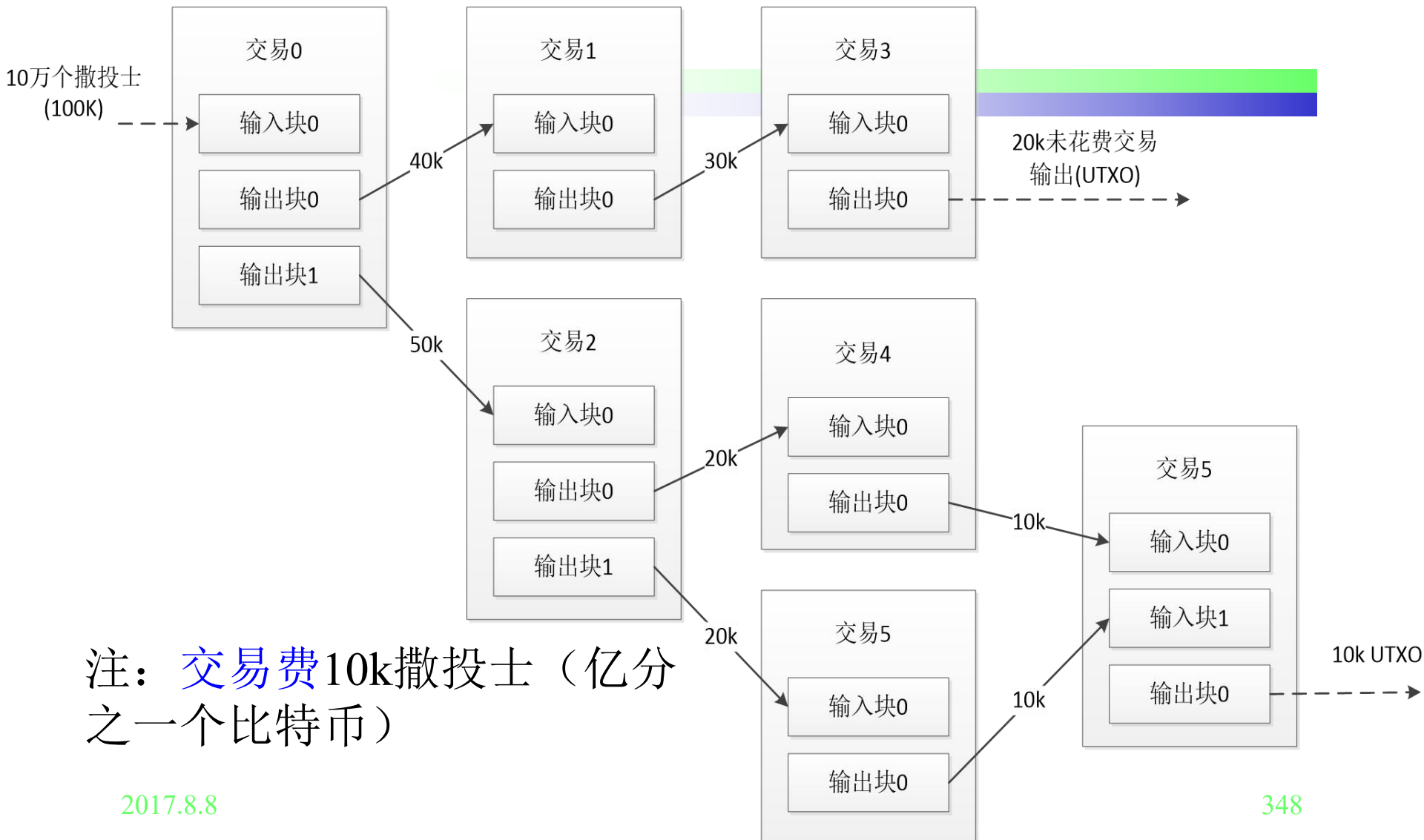
- 付给公钥哈希值（P2PKH）是一种标准的交易类型：利用接收方公钥作为地址
- 付给脚本哈希值（P2SH）交易是2012年提出的标准交易，使得支付方可以把赎回的脚本也包括在哈希值内。

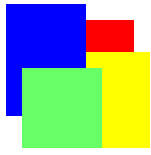


付给公钥哈希值（P2PKH）交易（A付给B）的示意图



# 一个典型交易的举例





# 工作证明：防止二次使用比特币

- 工作(量)证明（POW）是一个低于某个目标值的哈希值的尝试过程(淘金，也称挖矿)，它必须通过执行一定数量的计算才能获得。
- 比特币的工作证明充分利用了密码哈希算法中的随机特性，使得对于哈希值对应的数据有任何修改或者重新执行哈希运算，只会产生新的哈希值。
- 由于可能在不同计算时间，产生不同的哈希值，所以，最为困难的事情是找到低于某个值(目标值)的块首哈希值——不断修改“一次性数”，进行淘金



# 区块链应用的建议

- 应用区块链技术应该针对特定应用领域的去中心化信任管理需求，明确需要解决的问题。
- 在解决问题中，要灵活应用区块链综合的数据加密、真实性验证、对等网络的技术和实现方法，而不可拘泥于比特币系统中采用的区块链相关方法。
- 面向物联网的区块链应用，更加侧重于去中心化的数据可信管理，必须解决数据可信自动(必须限制可能的人工介入)采集、传递和存储的技术问题。



# 考试要求

## 一、耐心细致

仔细阅读题目，耐心梳理思路

## 二、诚实守信

诚实面对考试，遵守考场纪律

## 三、放松积极

放松身心，不要急躁；积极思考，不要放弃。



# 谢谢！

---

## 祝各位考出好成绩！