

计算机网络安全技术

2018 年 6 月 23 日

目录

| | |
|-----------------------------------|-----------|
| 1 基本密码学 | 3 |
| 1.1 基本概念 | 3 |
| 1.2 代换密码 | 3 |
| 1.3 置换密码 | 4 |
| 1.4 对称密码 | 4 |
| 1.4.1 Feistel 密码结构 | 4 |
| 1.4.2 常见对称密码 | 5 |
| 1.4.3 S-DES 算法 | 5 |
| 1.5 公钥密码 | 6 |
| 1.5.1 RSA 算法 | 6 |
| 1.5.2 Diffie-Hellman 算法 | 8 |
| 1.5.3 和对称密码的比较 | 9 |
| 1.6 密钥分配 | 9 |
| 1.6.1 传统对称密钥分配 | 9 |
| 1.6.2 非对称公钥发布 | 9 |
| 1.6.3 通过非对称公钥分配传统密钥 | 9 |
| 2 计算机网络安全体系结构 | 10 |
| 2.1 安全目标 | 10 |
| 2.2 手段 | 10 |
| 3 消息认证 | 10 |
| 3.1 攻击类型 | 10 |
| 3.2 消息认证基本概念 | 10 |
| 3.2.1 消息认证 | 10 |
| 3.2.2 基本框架 | 11 |
| 3.3 产生认证符 | 11 |
| 3.3.1 消息加密作为认证符 | 11 |
| 3.3.2 消息认证码 MAC | 11 |
| 3.3.3 消息哈希 | 11 |

| | |
|-----------------------------------|-----------|
| 3.4 哈希函数 | 11 |
| 3.4.1 哈希函数的理论要求 | 11 |
| 3.4.2 安全哈希函数的 Merkle 结构 | 12 |
| 3.4.3 常用哈希举例 | 12 |
| 3.5 数字签名 DSS 算法 | 12 |
| 4 访问控制 | 13 |
| 4.1 基本概念 | 13 |
| 4.2 访问控制模型 | 13 |
| 4.2.1 自主性访问控制 | 13 |
| 4.2.2 强制性访问控制 | 13 |
| 4.2.3 基于角色的访问控制 | 14 |
| 4.3 防火墙 | 14 |
| 4.3.1 设计目标 | 14 |
| 4.3.2 常用技术 | 14 |
| 4.3.3 访问控制列表 | 14 |
| 4.4 VLAN 虚拟局域网 | 15 |
| 5 IP 层级安全 | 15 |
| 5.1 IPsec | 15 |
| 5.1.1 SA | 15 |
| 5.2 IKE | 20 |
| 5.2.1 第一阶段 | 20 |
| 5.2.2 第二阶段 | 21 |
| 6 SSL | 21 |
| 6.1 特点 | 21 |
| 6.2 概念 | 21 |
| 6.3 协议 | 21 |
| 6.3.1 SSL 记录协议 | 21 |
| 6.3.2 握手协议 | 22 |
| 6.4 https 应用 | 22 |
| 6.4.1 http 的问题 | 22 |
| 6.4.2 https 基本概念 | 22 |
| 7 安全电子邮件 | 22 |
| 7.1 RFC 822 | 22 |
| 7.2 MIME | 22 |
| 7.3 请求响应协议 | 23 |
| 7.4 电子邮件安全问题 | 23 |
| 7.5 S/MIME | 23 |

| | |
|---------------------------|-----------|
| 8 安全电子商务 | 23 |
| 8.1 安全需求和安全问题 | 23 |
| 8.2 SET 协议 | 24 |
| 8.2.1 参与方 | 24 |
| 8.2.2 流程 | 25 |
| 9 入侵技术 | 27 |
| 9.1 入侵检测 | 27 |
| 9.1.1 基于统计的入侵检测 | 27 |
| 9.1.2 基于规则的入侵检测 | 27 |
| 9.1.3 蜜罐 | 27 |
| 9.2 软件入侵 | 27 |
| 9.2.1 后门 | 28 |
| 9.2.2 逻辑炸弹 | 28 |
| 9.2.3 特洛伊木马 | 28 |
| 9.2.4 Zombie | 28 |
| 9.2.5 病毒 | 28 |
| 9.2.6 蠕虫 | 28 |

1 基本密码学

1.1 基本概念

传统加密和私钥加密 传统加密包含代换, 置换密码及其组合, 重点依赖于算法的保密. 私钥加密亦称非对称加密, 算法和公约公开, 私钥保密.

块加密和流加密 输入是字符, 每次是处理一个字符还是处理一组字符.

无条件安全和计算安全 无条件安全即, 拥有无论多少密文对破译都没有帮助. 计算安全即, 破译代价大于加密数据本身价值, 或者破译时间长于加密数据有效时间.

记号

加密函数 $c = E_{K_E}(m)$, $m \in \mathcal{M}, c \in \mathcal{C}, K_E \in \mathcal{K}_E$

解密函数 $m = D_{K_D}(c)$, $c \in \mathcal{C}, m \in \mathcal{M}, K_D \in \mathcal{K}_D$

1.2 代换密码

如 Caesar 密码等单表代换密码, Playfair 密码和 Vigenere 密码等多表代换密码.

攻击 拥有足够的密文即可通过统计学分析破译.

1.3 置换密码

$D(m) = \sigma m$, 要求 $\mathcal{M} = \Sigma^L$, σ 是 $[0, L)$ 上的置换. 代换密码变换的是每个位置上的字母, 而置换密码变换的是消息中各个字母的位置.

攻击 频率分析, 包括多元组频率分析.

1.4 对称密码

1.4.1 Feistel 密码结构

Feistel 的密码观点

1. 使用乘积密码
2. 交替使用代换和置换

Shannon 的密码观点

扩散 密文不包含明文的统计信息, 每个明文字符影响多个密文字符

混淆 复杂的密文和密钥间的统计关系, 防止从密文推出密钥

Feistel 网络 大致如图所示.

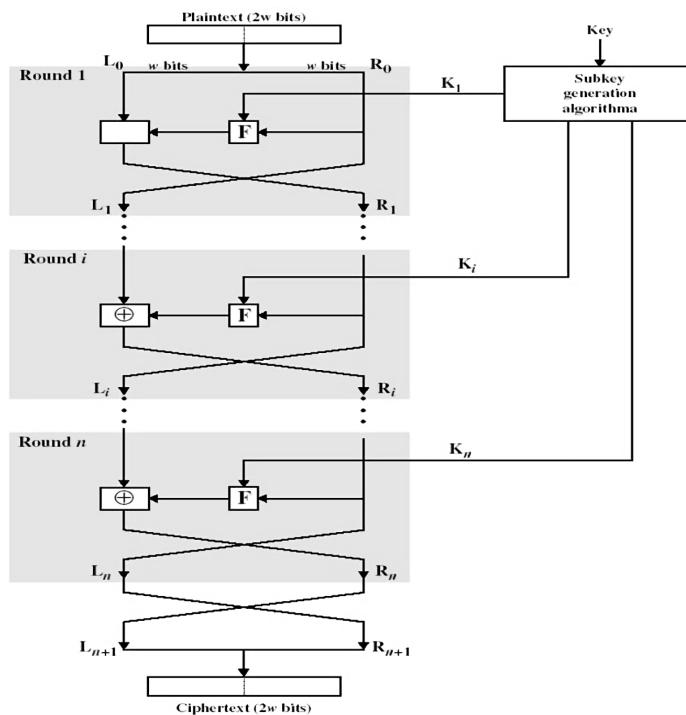


图 1: 经典的 Feistel 网络结构

Feistel 网络的元素

分组长度和密钥长度 越长越安全, 但是效率越低

迭代层数 越多越安全

K_i 的产生算法 越复杂越安全

F 函数 越复杂越安全

1.4.2 常见对称密码

对称密码的特点是只有一个密钥, 加密过程和解密过程是基本相同的.

通信双方如果需要交换密钥, 则需要在安全的信道上交换密钥.

对称密码的速度通常较非对称密码快, 因此常用于大量数据的加密上.

DES 算法 DES 是对称密钥算法, 基于 Feistel 网络结构. 密钥长度为 56 位, 块长度是 64 位, 迭代 16 轮.
和 Feistel 的区别是, 加密初始和末尾有一个置换.
已被破解.

3-DES 算法 密钥长度加倍到 112 位, 并且加密函数变成了

$$3 - \text{DES}(M) = \text{DES}(\text{DES}(\text{DES}(M)))$$

很安全, 但是效率很低.

Blowfish 算法 基于 Feistel 网络结构, 但是每轮中左右两半都进行计算.
未被破解.

RC5 算法 仍然是多轮加密, 但是每轮结构更加复杂.

AES 算法 AES 不是 Feistel 结构, 每一轮处理整个输入 (而非分成 2 部分).

1.4.3 S-DES 算法

此处的 S-DES 意为 Simplified DES, 主要做教学展示用. S-DES 基于 Feistel 结构, 输入输出是 8 位的, 密钥是 10 位的, 位指二进制位.

生成密钥 参考图 2. 其中置换如

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------------|---|---|---|---|---|---|---|---|---|---|
| $P10(n)$ | 2 | 4 | 1 | 6 | 3 | 9 | 0 | 8 | 7 | 5 |
| $shift_1(n)$ | 1 | 2 | 3 | 4 | 0 | 6 | 7 | 8 | 9 | 5 |
| $shift_2(n)$ | 2 | 3 | 4 | 0 | 1 | 7 | 8 | 9 | 5 | 6 |

另外函数 $P8(0, \dots, 9) = \langle 5, 2, 6, 3, 7, 4, 9, 8 \rangle$

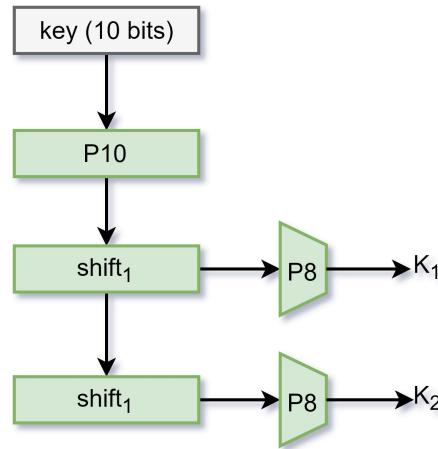


图 2: S-DES 生成密钥

加密 参考图 3 和 4, 其中有

$$IP(0, \dots, 7) = \langle 1, 5, 2, 0, 3, 7, 4, 6 \rangle$$

$$IP^{-1}(0, \dots, 7) = \langle 3, 0, 2, 4, 6, 1, 7, 5 \rangle$$

$$E/P(0, 1, 2, 3) = \langle 3, 0, 1, 2, 1, 2, 3, 0 \rangle$$

另外 S 的求法是, $S(n_0, n_1, n_2, n_3)$ 中, $n_0 * 2 + n_3$ 作为行, $n_1 * 2 + n_2$ 作为列, 寻找矩阵中对应元素

$$S_0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S_1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

1.5 公钥密码

公钥密码即非对称密码. 其中加密和解密使用的是不同的密钥, 称为公钥和私钥. 公钥公开, 用于加密和验证签名; 私钥保密, 用于解密和签名.

非对称密码的速度通常较对称密码要慢, 所以常常只用于签名¹, 或者用于加密发送对称密码的密钥.

加密方法 Alice 向 Bob 发送希望加密的信息, 则 Alice 使用 Bob 的公钥加密信息后发送给 Bob.

签名算法 Bob 希望验证 Alice 的身份, 其给出一个特定的消息, 要求 Alice 用 Alice 自己的私钥加密后发送给 Bob, 之后 Bob 再用 Alice 的公钥解密验证.

1.5.1 RSA 算法

原文和密文都是 $[0, N)$ 中的整数, 常常 N 是 2 的幂.

¹验证发送者身份的过程

Encryption Decryption

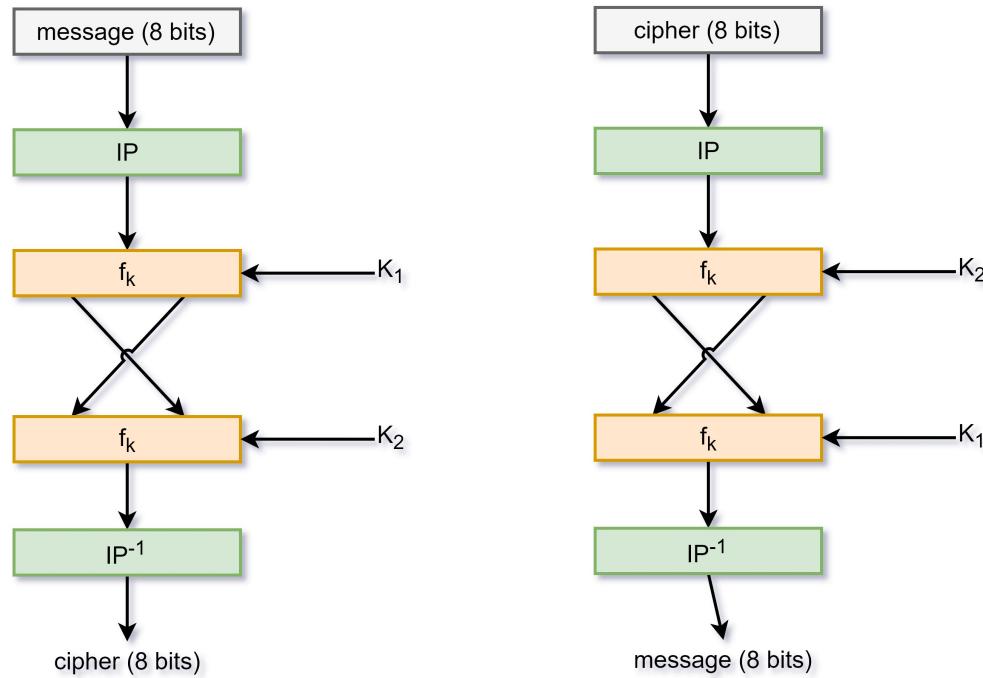


图 3: S-DES 加密

前置

1. 计算大素数 p, q .
2. 计算 $N = pq$, 公开.
3. 寻找一个小整数 e , $e < N$, $(e, N) = 1$
4. 求解 $de \equiv 1 \pmod{\varphi(N)} = (p-1)(q-1)$, 得到 d
5. 公钥为 $\langle e, N \rangle$, 私钥为 $\langle d, N \rangle$

加密 对于 $m \in [0, N]$, 加密函数如下

$$E(t) = m^e \pmod{N}$$

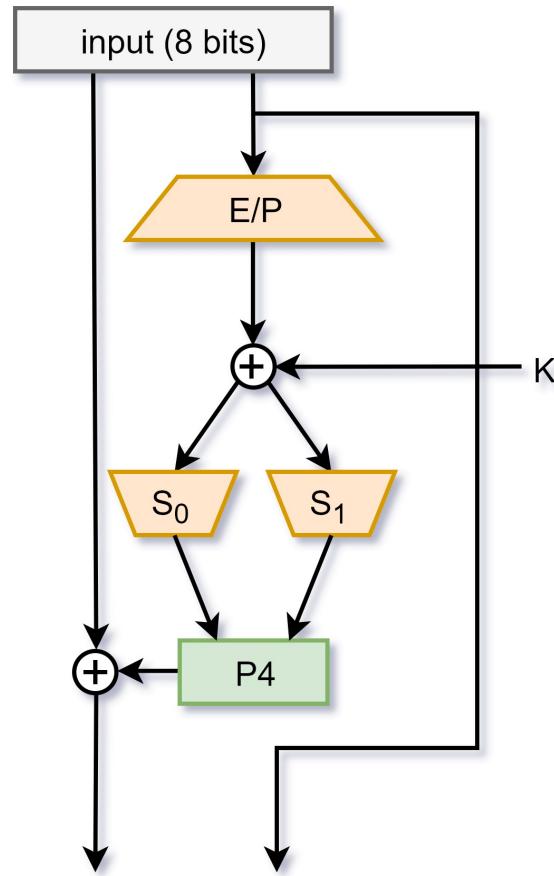
解密 对于 $c \in [0, N]$, 加密函数如下

$$D(c) = c^d \pmod{N}$$

原理 有效性由 Euler 定理保证

$$a^{\varphi(n)} \equiv 1 \pmod{n}, \quad \forall a : (a, n) = 1$$

安全性由大数分解困难性保证.

图 4: S-DES 的 F 函数

攻击

暴力破解 需要枚举 $t \in [0, N)$ 的空间

数学攻击 因子分解 N

计时攻击 按照加密时间推测私钥.

1.5.2 Diffie-Hellman 算法

用于密钥交换, 而非加密解密, 即通过双方自身的私有密文得到一个共有密文.

前置

- 寻找素数 p , 以及 \mathbb{Z}_p 的生成元 a i.e. a 是原根, 均公开.

私有密文产生共有密文

- 双方由自己的私有密文 X_A, X_B 计算公开的 $Y_A = a^{X_A} \bmod p, Y_B = a^{X_B} \bmod p$
- 双方由公开的 Y_A, Y_B 计算得到共有密文 $K = Y_B^{X_A} \bmod p = Y_A^{X_B} \bmod p = a^{X_A X_B} \bmod p$

原理 安全性由离散对数困难性保证.

1.5.3 和对称密码的比较

- 笼统地说, 非对称密码并不比对称密码安全.
- 非对称密码不是通用的, 对称密码仍然由于快速而广泛使用
- 公钥密码的密钥分配不必传统密钥分配简单

1.6 密钥分配

1.6.1 传统对称密钥分配

可能的方法

1. Alice 选择密钥, 亲自交给 Bob
2. 第三方 Charlie 选择密钥, 亲自交给 Alice 和 Bob
3. Alice 选择密钥, 用最近使用的密钥加密后发给 Bob
4. 第三方 Charlie 选择密钥, 通过某个秘密渠道交给 Alice 和 Bob

密钥分发中心 基本假设: 每个人有一个仅他自己和 KDC 知道的主密钥. 此假设下, Alice 和 Bob 得到一次性次密钥的方法:

1. Alice 向 KDC 请求次密钥
2. KDC 发送给 Alice 消息, 消息用 PK_a 加密, 包含用次密钥 K_s , 以及 PK_b 加密的 K_s 和 ID_a
3. Alice 将 PK_b 加密的消息发送给 Bob

为了效率和安全性, KDC 通常也是层次性的, 而非一个巨大的中心 KDC.

1.6.2 非对称公钥发布

公开发布 Alice 向所有人广播自己的公钥. 问题是容易伪造广播消息.

公开目录 只有一个受信任的实体 (管理员) 能够广播公钥. 问题是如果受信任的实体被攻击, 公钥可以任意更改.

公钥授权 双方通讯之前先向管理员请求对方公钥, 管理员返回消息时用管理员私钥加密.

1.6.3 通过非对称公钥分配传统密钥

非对称加密的问题是效率太低, 一般通过非对称加密传输传统密钥, 之后消息用传统加密方法求解.

Merkel 朴素方法 Alice 给 Bob 请求共有密钥. Bob 用 Alice 公钥加密某密文后传输给 Alice, Alice 之后使用自己的私钥解密得到共有密文. 问题: 中间人攻击, 考虑若 Bob 是假 Bob.

保密真实方法

1. Alice 发送给 Bob: $E_{K_{B, pub}}(ID_A, N_A)$, N_A 是一个随机校验数
2. Bob 得到 ID_A, N_A , 发送给 Alice: $E_{K_{A, pub}}(N_A, N_B)$
3. Alice 得到 N'_A, N_B , 检查 $N_A = N'_A$, 发送给 BE _{$K_{B, pub}$} (N_B)
4. Bob 得到 N'_B , 检查 $N_B = n'_B$

2 计算机网络安全体系结构

2.1 安全目标

保密性 Confidentiality 未授权的实体不能获得信息内容

完整性 Integrity 信息不能被篡改, 或者能检测篡改

可用性 Availability 授权用户能够访问资源, 防止 DoS 攻击

2.2 手段

加密 防止未授权的外人获得通信内容, 如防止窃听.

认证 验证通信对等的身份真实性, 如防止中间人攻击. 可以通俗的理解为, “发送这条消息的人的确是 Alice”.

数字签名 相当于对通信对等体和消息同时认证, 如防止抵赖. 可以通俗的理解为, “Alice 的确发送过这条消息”.

3 消息认证

3.1 攻击类型

泄密 保密性范畴.

传输分析 保密性范畴.

伪装 消息认证.

内容修改 消息认证.

抵赖 数字签名, 以及协议设计.

3.2 消息认证基本概念

3.2.1 消息认证

确认发送方是真实的, 确认消息违背篡改.

消息认证就是验证所收到的消息确实是来自真正的发送方且未被修改的消息

3.2.2 基本框架

发送方产生一个认证符; 接收方产生一个认证符, 并且检查双方认证符是否匹配.

3.3 产生认证符

3.3.1 消息加密作为认证符

对称加密 为了认证, 要么消息空间是稀疏的, 要么消息中带有校验符号 FCS. FCS 的计算是在加密之前, 将发送的内容计算出一个校验符, 附到发送内容后一并加密发送.

可以提供加密和认证, 但是无法提供数字签名.

非对称加密 使用私钥加密可以完成签名和认证, 使用公钥加密可以提供保密性, 两者都使用可以同时提供加密和认证以及签名.

但是这样需要 4 次加解密运算, 代价较高.

3.3.2 消息认证码 MAC

通过消息和密钥 (两者都需要, 以验证这条消息是发自这个发送者), 利用公开的算法计算消息认证码 $MAC = C_K(M)$, 其中 K 是共有密钥, M 是消息, C_K 生成一个固定长度的短数据块.

提供认证, 但是不提供数字签名: 接收方也有 K , 因此可以伪造消息.

可以先加密在使用 MAC, 提供加密和认证.

3.3.3 消息哈希

通过消息, 计算认证符 $MD = \text{Hash}(M)$, M 是消息, **Hash** 输出固定长度的短数据. 亦称消息摘要.

哈希本身不包含对于发送者的验证, 因此一定要对哈希加密. 根据需要的服务, 具体有

- 对称加密中, 可以加密消息和哈希码, 可以完成加密和认证.
- 对称加密中, 可以只加密哈希码, 可以完成认证.
- 非对称加密中, 用发送方的私钥加密哈希码, 可以完成认证和数字签名.
- 非对称加密中, 用发送方的私钥加密消息和哈希码, 可以完成加密, 认证和数字签名.

3.4 哈希函数

3.4.1 哈希函数的理论要求

要求哈希函数 $h = H(M)$ 有如下性质

单向性 对于 h , 难以寻找 M , 满足 $H(M) = h$

弱抗碰撞 对于 M , 难以寻找 M' , 满足 $H(M) = H(M')$

强抗碰撞 $\forall M$, 难以寻找 M' , 满足 $H(M) = H(M')$

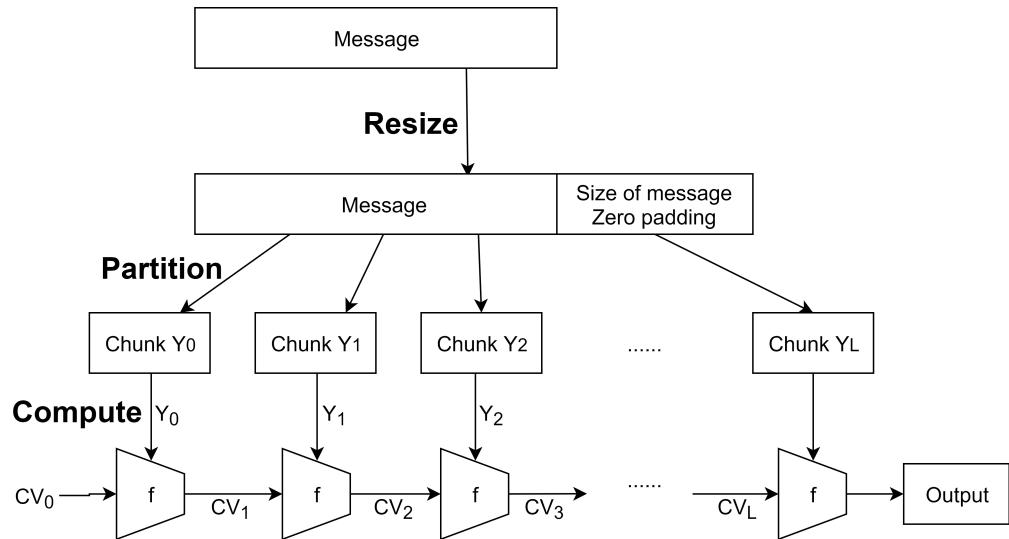


图 5: Merkle 哈希结构

3.4.2 安全哈希函数的 Merkle 结构

参见图 5.

3.4.3 常用哈希举例

有 MD5, SHA-1, SHA-2, GOST 等等.

MD5 有如下特点

- 消息可以无限长
- 采用小端结构, 将消息表示为 32 位字的序列
- 填充部分包含了消息的长度
- 填充后消息位数是 512 的倍数, 每个分组 Y_i 大小为 512 位 (64 字节, 16 个字)
- 中间结果 CV_i 和最后结果 MD 的长度是 128 位 (16 字节, 4 个字寄存器)
- 由四轮运算组成, 每轮 16 步迭代

3.5 数字签名 DSS 算法

应用场景 来自消息发送双方的攻击, 如接收方伪造发送, 发送方抵赖发送.

分类 直接数字签名 (仅通信双方), 仲裁数字签名 (受信任的实体).

前置

全局公钥 p 是素数

q 是素数且 $q \mid p - 1$

$$g = h^{(p-1)/q} \bmod p$$

用户私钥 x , 满足 $x < q - 1$

用户公钥 $y = g^x \bmod p$

随机数 $k < q$, 每次认证都应产生一个新的 k , 且 k 保密

DSS 算法签名 签名用一组数 (r, s) 代表, 计算过程如下方程, 哈希函数 \mathbf{H} 一般取 SHA-1

$$\begin{aligned} r &= g^k \bmod p \bmod q \\ s &= (k^{-1}(\mathbf{H}(M) + xr)) \bmod q \end{aligned}$$

验证要求

$$g^{(s^{-1}M) \bmod q} y^{(s^{-1}r) \bmod q} = r$$

4 访问控制

4.1 基本概念

主体 提出访问资源请求的人/实体.

客体 含有被访问资源的实体.

访问 对资源的使用行为, 有时还需包括主客体.

访问控制矩阵 定义不同主体对于不同客体可以执行的操作, 如

| | O_2 | O_3 | O_4 |
|-------|-------|-------|-------|
| S_1 | R/W | R | |
| S_2 | W | | |
| S_3 | R/W | R/W | R/W |

按列看, 就是客体的访问控制列表; 按行看, 就是主体的访问能力列表.

4.2 访问控制模型

4.2.1 自主性访问控制

每个客体有一个管理者, 管理者将访问权限授权给其他人. 可以看成以主体为核心. 如 Unix 文件系统.

性质 易用方便灵活. 但是不能控制信息的流动, i.e. 其他人取得资源后可以自由分发.

4.2.2 强制性访问控制

每个主体和客体有固定的安全级别, 由系统管理员决定. 可以看成以客体为核心.

No Read Up 主体只能读取安全级别更低的客体.

No Write Down 主题只能写入安全级别更高的客体.

4.2.3 基于角色的访问控制

主体 (用户) 属于用户组 (角色). 每个角色 (而非主体) 对于不同的客体有不同访问权限. 可以看成以访问为核心.

4.3 防火墙

在两个不同网络间建立访问控制的软硬件.

4.3.1 设计目标

监控两个网络间所有通信流量, 只有授权的流量才允许通过.

4.3.2 常用技术

基于控制, 有

服务控制 基于服务端口号 / IP 地址

方向控制 内到外 / 外到内

用户控制 基于用户的身份, 如 VPN

行为控制 基于用户的行为进行控制, 如垃圾邮件过滤

基于分层, 有

网络层 包过滤技术.

根据定义好的过滤规则, 包含 IP 地址, 端口和协议等, 审查每个数据包. 性能较高, 但控制能力不强.
如路由器中的 ACL.

网络层 地址转换.

完成转发和地址转换. 不提供额外的安全性, 但是可以隐蔽内部网络, 节省地址空间

传输层 电路层网关.

检查包所属的会话, 即是不是属于客户端和服务器的某一个链接, 不检查协议和内容. 不支持无连接的 UDP, 性能开销较大.

应用层 应用层代理.

功能强大, 但是性能较低, 并且实现麻烦.

4.3.3 访问控制列表

一组预先定义好的规则. 其根据包头, 指定包是否被拦截.

标准和拓展 标准 ACL 只检查 Frame 中源 IP 地址. 拓展 ACL 还支持端口, 目标 IP, 协议, 检查 Frame 和 Packet.

路由过程中, 路由前后分别有入口/出口端 ACL. 入口端 ACL 在查路由表前检查, 出口端 ACL 在转发前检查.

4.4 VLAN 虚拟局域网

基本概念 VLAN 类似一个独立的网桥, 但是可能一个 VLAN 跨越不同的交换机, 同一个交换机有不同的 VLAN.

类型 如

基于物理接口 按照交换机的接口分配.

配置简单, 但是不灵活.

基于 MAC 地址 局域网中每个 MAC 地址有对应的 VLAN 分组.

基于协议 根据网络层协议分划 VLAN.

优点

- 解决人员维护性
- 控制广播流量, 防止交换机后向学习的洪泛造成的“洪范灾难”
- 增强安全性

缺点 缺乏标准

5 IP 层级安全

5.1 IPsec

与 IP 在同一层. 功能

认证 确认包的发送者真实, 包不被篡改

加密 流量管理

密钥管理 密钥的安全交换

5.1.1 SA

背景 假设 Alice 希望给 Bob 发送数据. 无论加密(对称加密)还是认证(HMAC²), 他们都需要一个共有的密钥. IPsec 讨论中假设这个双方已有这个共有密钥了.

SPD 安全策略数据库, 其过滤所有 IP 流量. 并且对于每个 IP 包, 其指定对于此 IP 包的处理, 是丢弃DISCARD, 直接通过IP发送BYPASS, 还是通过IPsec即PROTECT. 如果指定是PROTECT, 它还需指定是ESP还是AH, 使用传输模式还是隧道模式, 之后或者利用IKE生成一个SA记录到SAD中, 或者交由SAD处理.

²即 MAC, 参考 rfc 2104

SA 内容 Bob 可能有很多个人境连接都使用 IPsec, 因此每个连接都需要一个标识, 用来让 Bob 确定这个连接的参数, 如哈希算法, 公共密钥等. IPsec 中此标识实现为 SPI, 是一个 32 位的整数.

Bob 本地有一个 SAD, 对于每一个有效的 SPI 在 SAD 中都有对应的 SA (包含 SPI, 使用 ESP 还是 AH, 目标 IP 地址), 以及源地址, SA 记录等等.

传输模式和隧道模式 对于加密或认证, 如果针对的是 IP 载荷 (即 TCP 头加载荷), 则成为传输模式, 用于端到端的传输. 通常就在主机上实现. 如果针对的整个 IP 包, 则成为隧道模式, 用于网络上构建 VPN. 通常在防火墙或者路由器上实现.

原理 IPsec 有两种协议, 有 AH, 用于认证; 以及 ESP, 用于加密和可选的认证. 参见如图

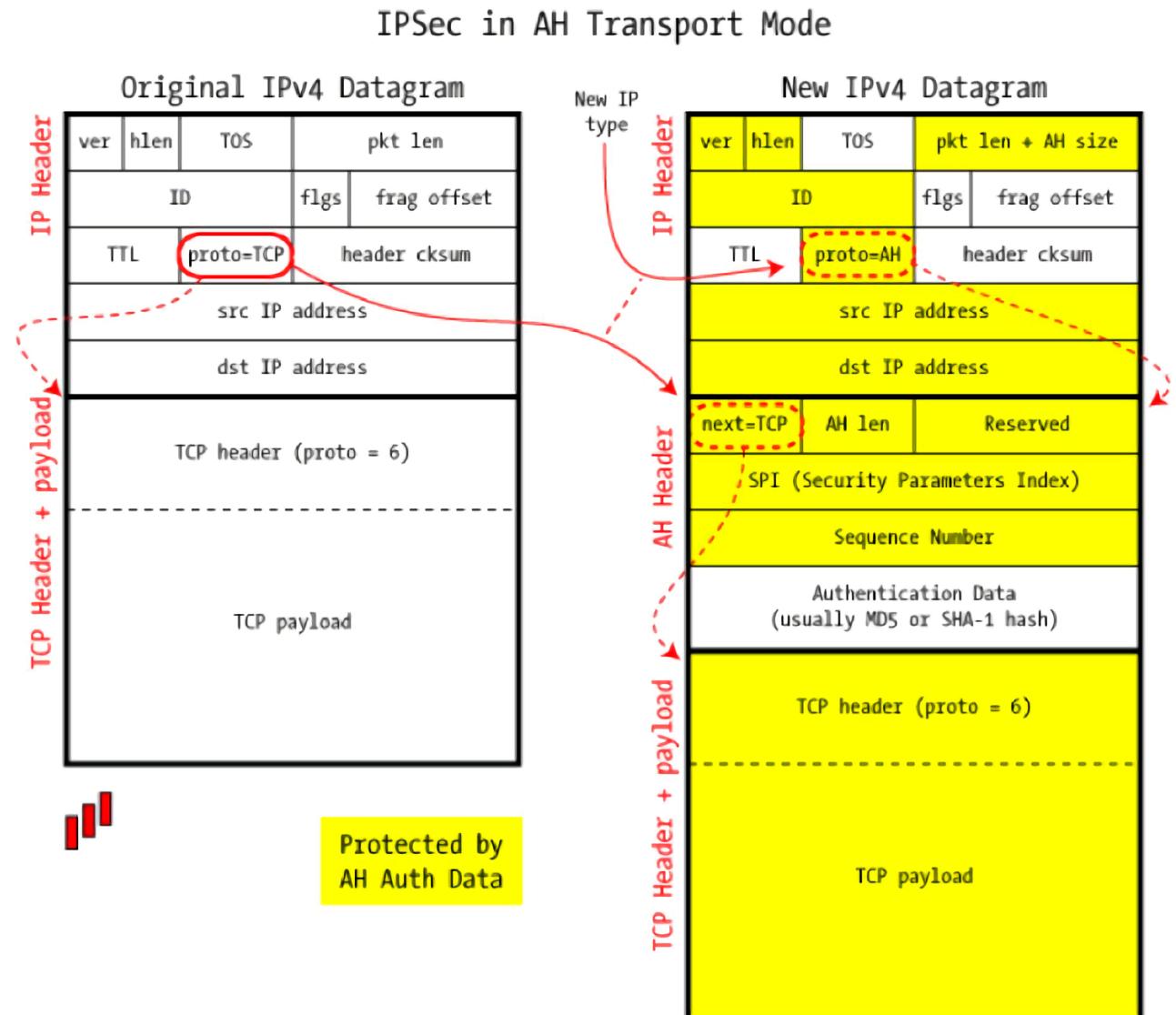


图 6: 传输模式的 AH

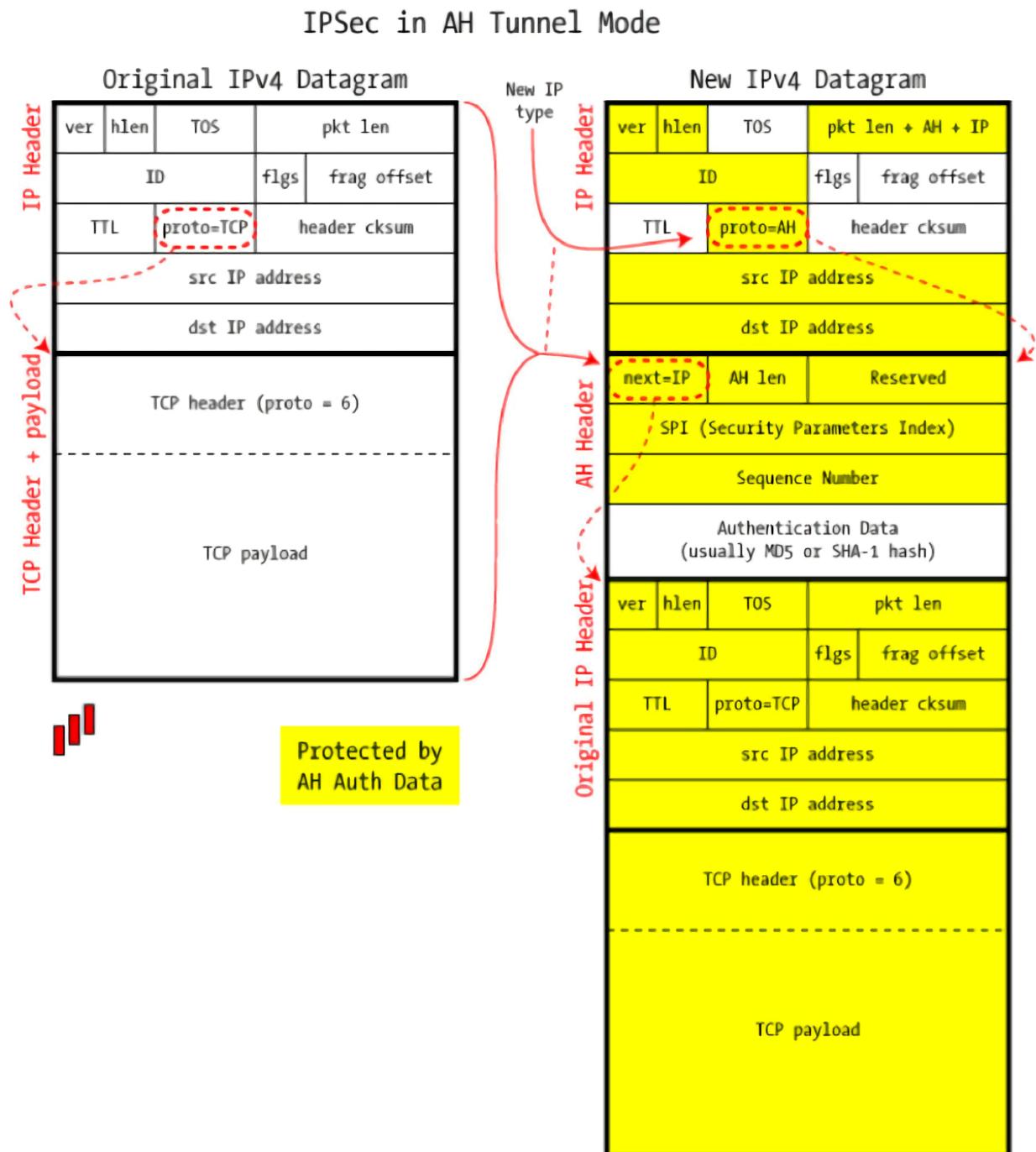


图 7: 隧道模式的 AH

IPSec in ESP Transport Mode

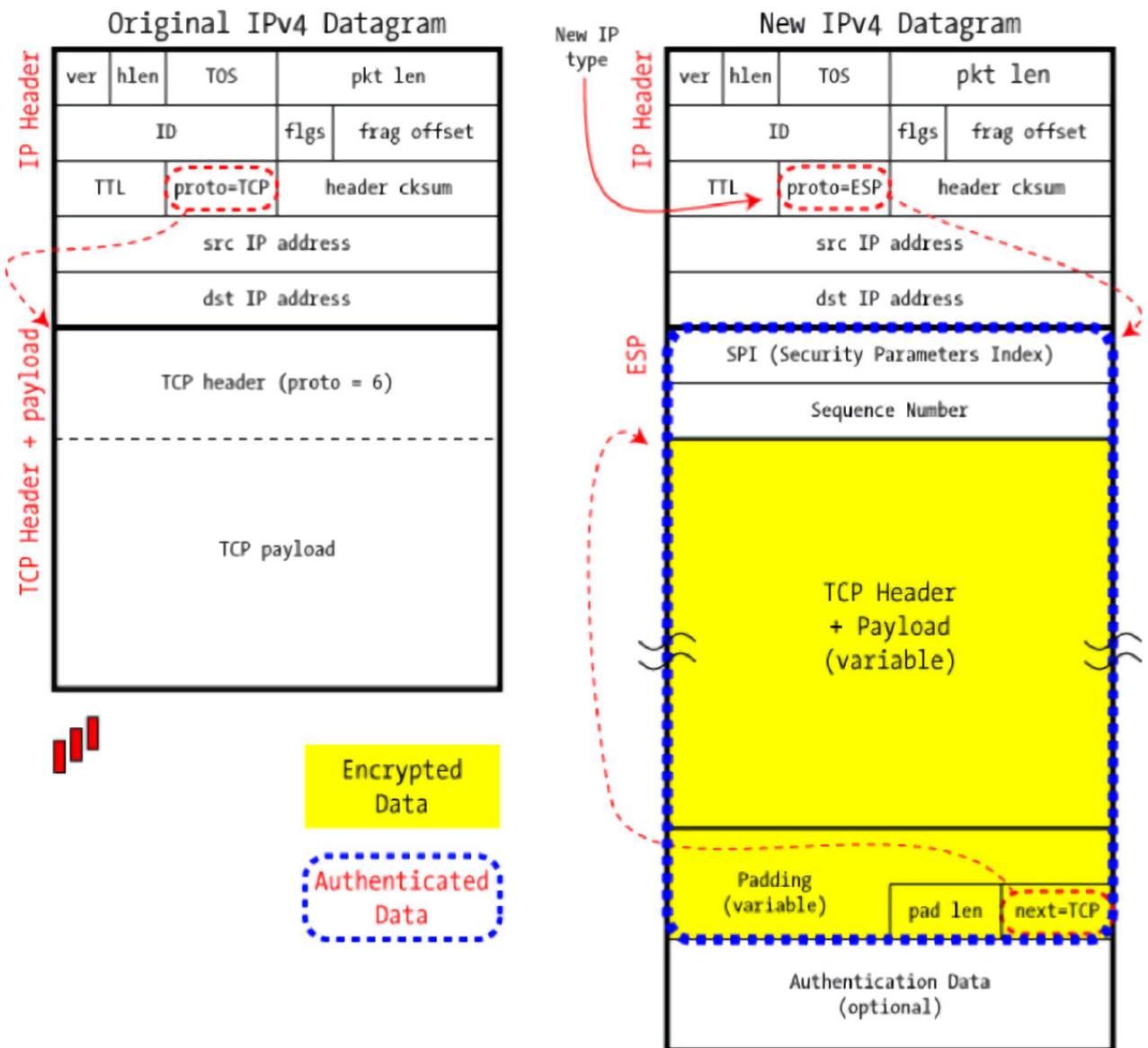


图 8: 传输模式的 ESP

IPSec in ESP Tunnel Mode

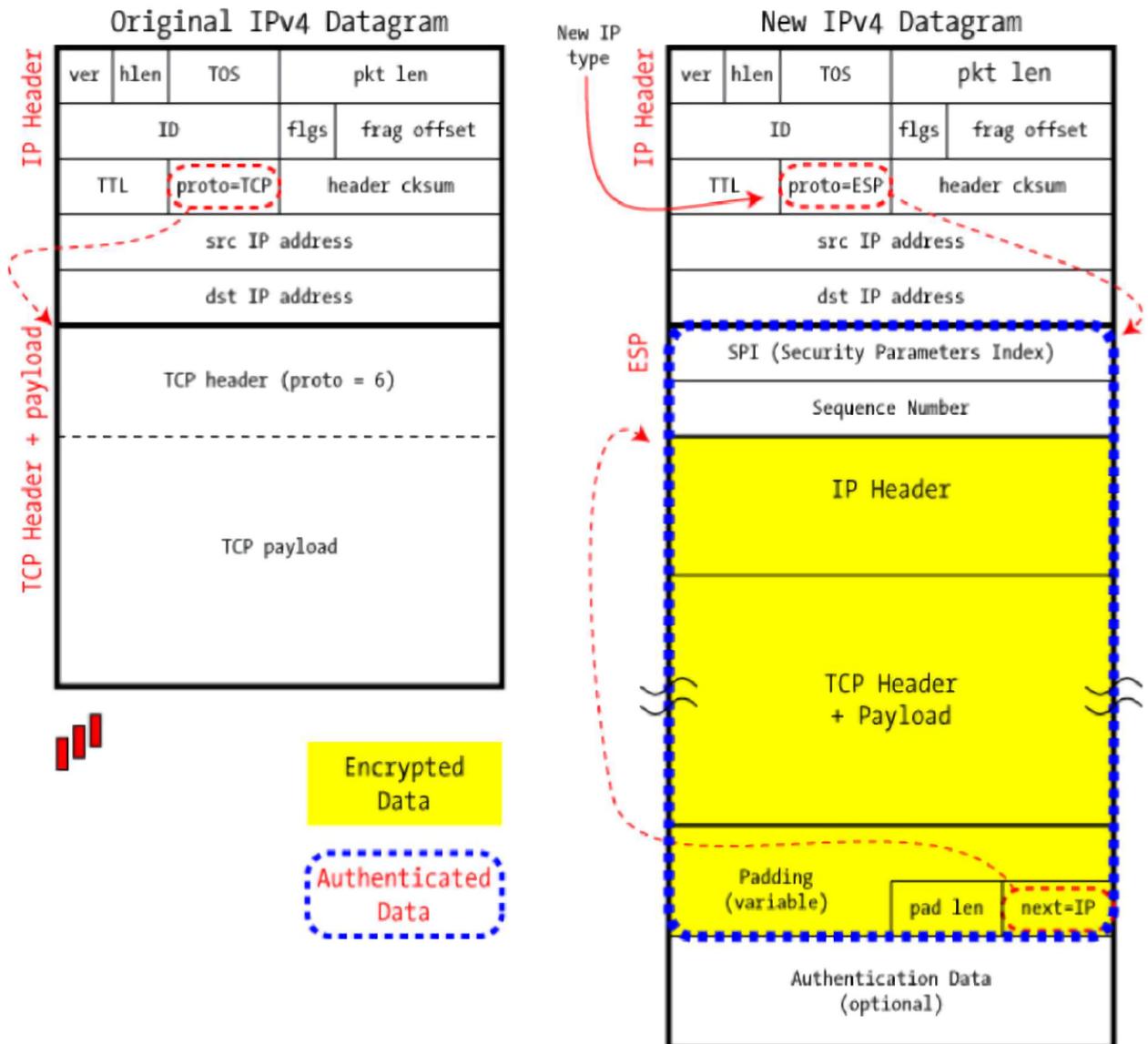


图 9: 隧道模式的 ESP

5.2 IKE

IKE 位于应用层 (UDP 端口 500), 目的是在不安全的互联网上交换密钥, 使得通信双方能得到一个共有的公共密钥, 用于对称加密或签名. 此外 IKE 还需要和 SPD, SAD 协同工作, 后两者可能请求 IKE 协商 SA.

可以采用传输模式端对端地工作, 也可以利用隧道模式在安全网关之间工作.

朴素 DH 的问题 朴素 DH 容易受到中间人攻击, 即欺骗者对 Alice 声称自己是 Bob, 对 Bob 声称自己是 Alice. 解决需要提供 Alice 和 Bob 的身份验证.

5.2.1 第一阶段

此阶段协商 IKE SA. 有主模式main mode 和快速模式aggressive mode.

双方认证可以有多种方法, 此处以签名认证为例, 摘抄 rfc 如下

```
Phase 1 is where the two ISAKMP peers establish a secure,
authenticated channel with which to communicate.
```

5.1 IKE Phase 1 Authenticated With Signatures

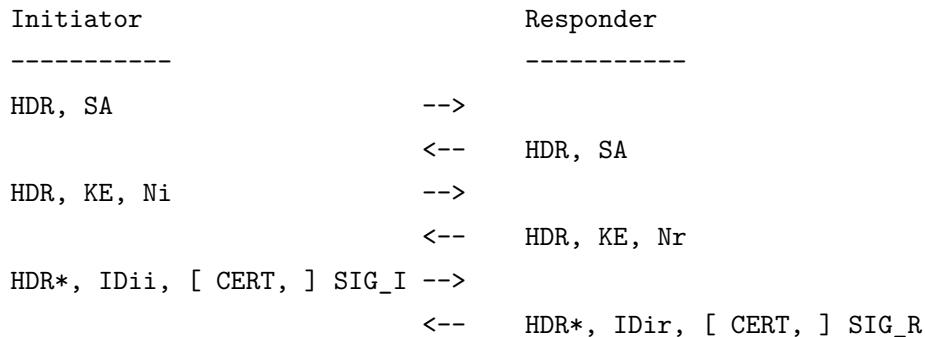
Main Mode with signature authentication is described as follows.

The first two messages negotiate policy.

The next two exchange Diffie-Hellman public values

and ancillary data (e.g. nonces) necessary for the exchange;

The last two messages authenticate the Diffie-Hellman Exchange.



Aggressive mode with signatures in conjunction with ISAKMP is described as follows:

The first two messages negotiate policy, exchange Diffie-Hellman public values
and ancillary data necessary for the exchange, and identities.

In addition the second message authenticates the responder.

The third message authenticates the initiator
and provides a proof of participation in the exchange.



```

<--      HDR, SA, KE, Nr, IDir,
          [ CERT, ] SIG_R
HDR, [ CERT, ] SIG_I      -->

```

In both modes, the signed data, SIG_I or SIG_R, is the result of the negotiated digital signature algorithm applied to HASH_I or HASH_R respectively.

c.f.

HDR: ISAKMP header; HDR* denotes payloads encrypted.
 SA: an SA negotiation payload with one or more proposals.
 KE: public information exchanged in a Diffie-Hellman exchange.
 Nx: nonce payload;
 IDx: identification payload, e.g. IP address
 SIG_x: necessary info ($g^x y$, SAx, IDx, Nx etc) signed by x.

5.2.2 第二阶段

协商如 IPsec 的加密算法, 哈希算法, DH 组等等参数. 使用快速模式 Quick mode, 需要三次消息交换.

6 SSL

6.1 特点

- 在 IP/TCP 参考模型中, SSL / TLS 位于应用层和传输层之间.
- SSL 工作在 TCP 之上, 不支持 UDP.

6.2 概念

会话 交流双方的一个虚拟的连接关系, 指定了密码算法, 主密钥等等消息.

连接 一个通信信道, 通常是一个 TCP 连接.
一个会话可以被多个连接先后使用.

6.3 协议

6.3.1 SSL 记录协议

SSL 记录协议定义了 SSL 传输信息的格式, 是其他 SSL 协议的基础.
高层数据, 通过 SSL, 会经过一下步骤

1. 分段, 将高层数据分成小块
2. 压缩, 这一步是可选的
3. 计算 MAC 附在数据之后

4. 使用共享密钥对数据对称加密
5. 在加密后的数据前添加 SSL 头部

6.3.2 握手协议

握手协议有以下功能

身份认证 认证服务器的身份，并可选地认证客户的身份

协商密码算法 包含对称密码算法, MAC 算法以及密钥交换算法

协商主密钥 即 master secret, 不过其并不直接用于加密和认证

之后分为 4 个阶段

1. 建立连接, 协商算法
2. 服务器认证, 服务器发送密钥交换相关消息
3. 可选的客户端认证, 客户端发送密钥交换相关消息
4. 完成确认

6.4 https 应用

6.4.1 http 的问题

嗅探监听 信息不加密

篡改 信息没有认证

伪造服务器 http 不验证服务器的可信度

6.4.2 https 基本概念

相对于 http, https 将消息按照 SSL 加密后再传输, 并且支持通过服务器证书完成的身份验证等.

7 安全电子邮件

7.1 RFC 822

定义了最基础的电子邮件传输方式.

消息格式 消息包含信封和内容两个部分, 通过空行隔开. 要求消息是 ASCII 文本消息.

信封的格式如“关键字: 参数”, 通常有From, To, Subject, Date 等关键字.

7.2 MIME

针对 RFC 822 在多媒体传输, 编码等问题, 提出的扩展.

在报头部分增加若干关键字如Content-Type, 并且允许多种编码如base64.

7.3 请求响应协议

RFC 821 即 SMTP 协议, 只处理 ASCII 数据传输. 没有任何安全措施, 没有密码登陆.

POP3 类似 SMTP 协议, 允许用户名密码登陆.

IMAP4 允许有选择地接收邮件等功能.

7.4 电子邮件安全问题

匿名转发 收件人无法知道真正的发件人

邮件伪造 假冒其他人发送邮件

抵赖 可伪造的都可以抵赖

其他 垃圾邮件和邮件炸弹, 包括邮件病毒等等

7.5 S/MIME

允许发送 MIME 数据, 提供认证, 加密, 数据完整性和抗抵赖.

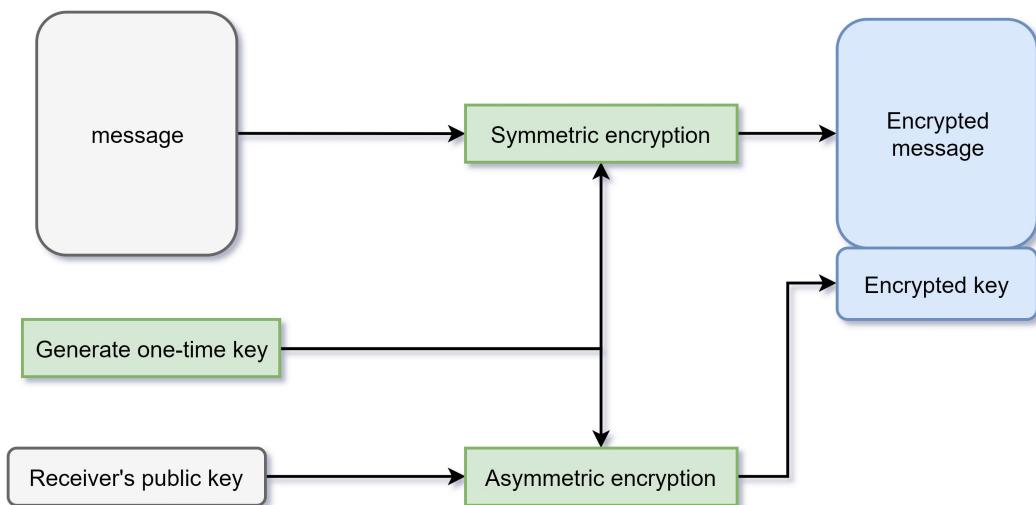


图 10: S/MIME 的加密过程

S/MIME 支持透明签名, 即对空消息进行数字签名后发送.

8 安全电子商务

8.1 安全需求和安全问题

- 资金流动的保密性. 防止第三方截获交易信息
- 支付结算数据的完整性. 防止篡改数据
- 支付双方身份认证.

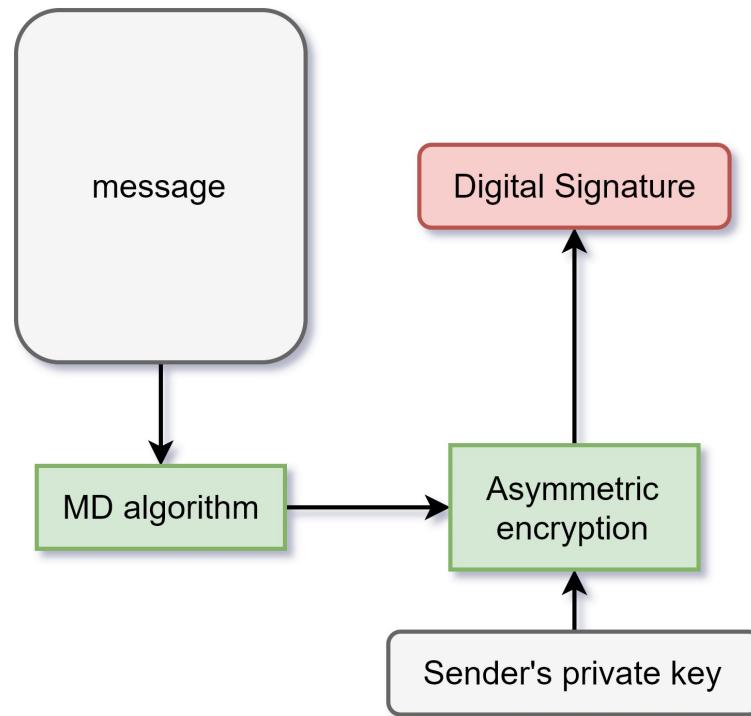


图 11: S/MIME 的数字签名

- 抗抵赖
- 效率等

SET 协议基于信用卡, 完成了

- 私密性
- 保密性
- 完整性
- 抗抵赖

8.2 SET 协议

8.2.1 参与方

持卡人 意即消费者

商家 事先和收款行建立信任关系

发卡行 消费者持卡的银行

收款行 代替商家与多个发卡行联系, 验证持卡人信息的有效性

证书权威 可信的第三方, 为持卡人, 商家, 收款行提供证书

支付网关 收款行控制, 处理商家的支付报文.

8.2.2 流程

双签名 交易中有两种信息, 订单相关和支付相关的. 只有商家才能看到订单相关的, 只有银行才能看到支付相关的. 然而需要对两种信息都签名, 并允许商家和银行验证两种信息的签名.

双签名的计算即如

$$\text{Dual Signature} = E_{KR_C}[H(\text{PIMD} \parallel \text{OIMD})]$$

购买请求 如图

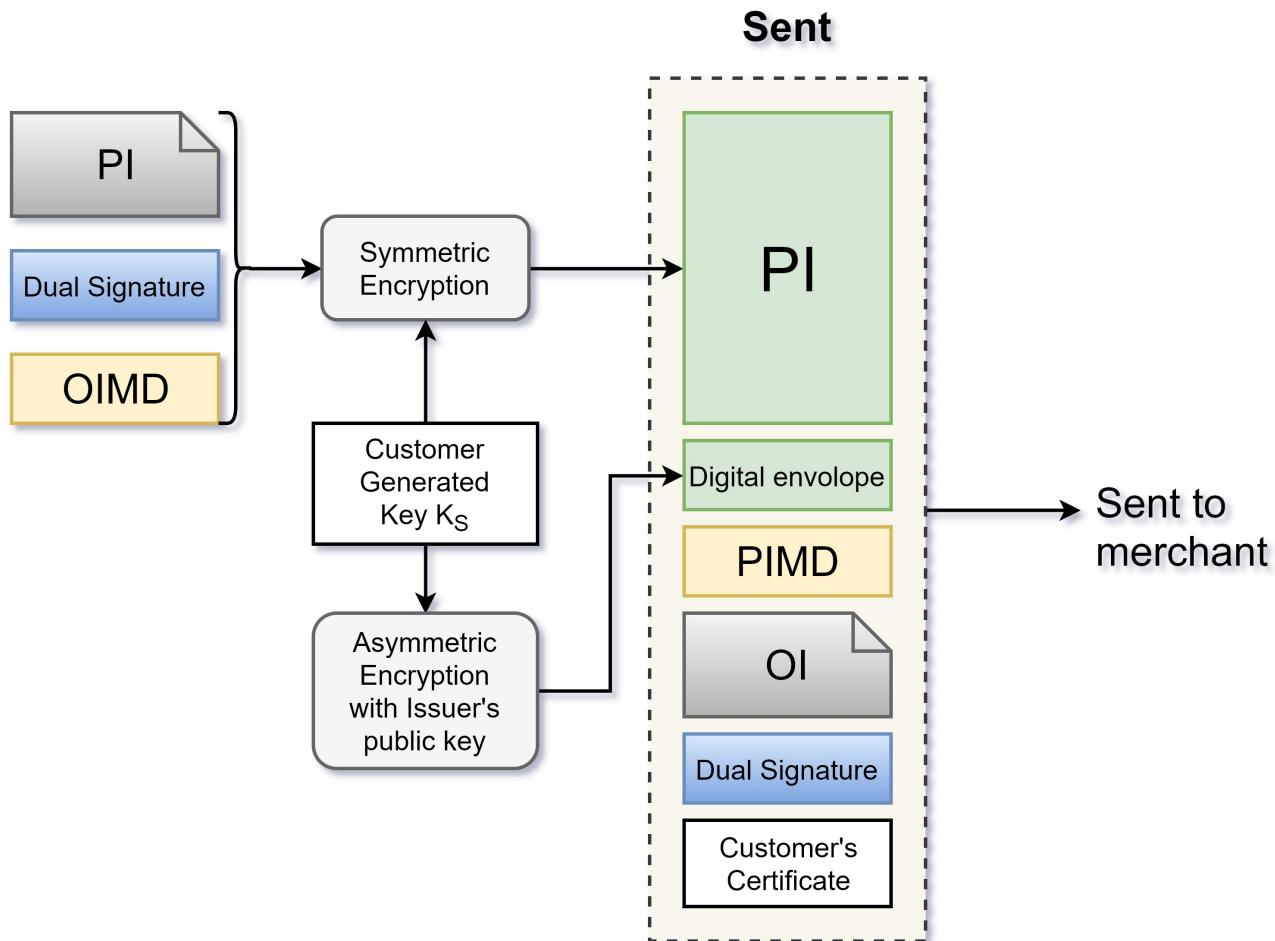


图 12: 购买请求, Customer 端

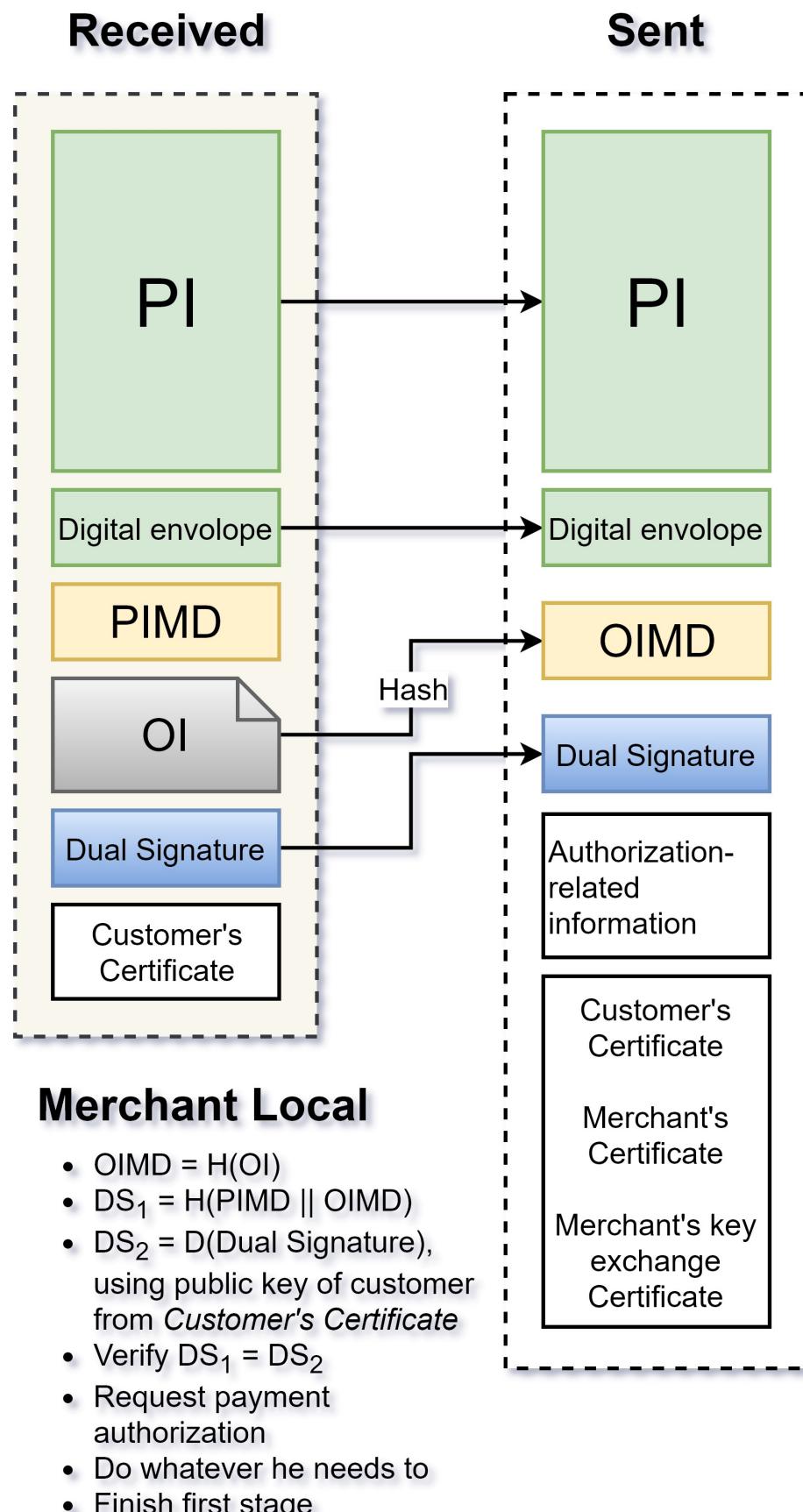


图 13: 购买请求, Merchant 端

支付授权

1. Merchant 给 Acquirer 发送支付授权请求
2. Acquirer 确认授权, 给 Issuer 发送划款请求

支付获取 完成转账业务

9 入侵技术

9.1 入侵检测

审计记录 包含用户活动记录, 用于检测用户的行为.

9.1.1 基于统计的入侵检测

能够防范假冒者, 但是不能防范合法用户.

阈值检测 检测一段时间内某用户产生的各种事件. 如果, 事件发生次数超过其阈值, 则认为可能存在入侵.
本身是很粗糙的, 容易误判.

轮廓检测 为每个用户建立一个行为轮廓, 学习其行为模式, 检测用户行为模式的异常变化.

9.1.2 基于规则的入侵检测

基本原理是检测系统中发生的事件, 运用先定的规则集确定某一个活动模式是否可疑. 其中先定的规则集是专家系统定义的.

9.1.3 蜜罐

创建一个正常用户不会访问的蜜罐系统, 引诱攻击者攻击这个没有有用信息的蜜罐系统.
任何对于蜜罐的访问都是可疑的.

9.2 软件入侵

入侵技术分为用户入侵和软件入侵, 课程中着重考虑软件入侵. 软件入侵主要考虑恶意软件.

| | 病毒 | 蠕虫 | 木马 |
|------|------------|-------|-----------|
| 宿主 | 需要 | 不需要 | 需要 |
| 表现形式 | 不以文件 | 独立的文件 | 伪装成其他文件 |
| 传播方式 | 依赖宿主 | 自主传播 | 依靠用户主动传播 |
| 危害 | 破坏数据/系统完整性 | 侵占资源 | 留下后门 |
| 传播速度 | 快 | 很快 | 慢(不能自我复制) |

表 1: 各种恶意软件的特点

9.2.1 后门

软件中秘密入口, 可以绕过通常步骤获得权限. 控制方式包含本地权限提升, 远程执行程序等.

9.2.2 逻辑炸弹

当特定事件发生时, 在被执行制造破坏的代码.

9.2.3 特洛伊木马

伪装成正常程序, 欺骗安装和运行, 使攻击者能获得权限. 类似后门, 但重点在伪装成正常程序的欺骗性.

9.2.4 Zombie

秘密接管 Internet 上的计算机, 利用这些计算机发送攻击 (常是 DoS).

9.2.5 病毒

一段可以通过修改自身, 修改其他程序的程序片段.

9.2.6 蠕虫

完整独立的计算机程序, 能够自我复制, 并通过计算机网络传播.