

第9章 电子邮件安全 (Email Security)

1. 电子邮件安全概述

电子邮件 (Email) 是互联网最广泛的应用，但基于 SMTP 协议的默认传输机制存在安全隐患。本章核心解决两个问题：身份认证 (**Authentication**) 与保密性 (**Confidentiality**)。

1.1 安全目标

- 保密性 (**Confidentiality**): 防止邮件内容在传输和存储中被非授权窃听。
- 身份认证 (**Authentication**): 确认发件人确实是其声称的身份 (防止伪造)。
- 消息完整性 (**Message Integrity**): 确保邮件内容在传输中未被篡改。
- 源的不可否认性 (**Non-repudiation of origin**): 发件人无法抵赖曾发送过该邮件。

2. PGP (Pretty Good Privacy)

2.1 概述

PGP 是由 Phil Zimmermann 开发的综合性安全工具，提供电子邮件和文件存储的保密与认证服务。

- 定位：偏向个人用户，去中心化。
- 特点：不依赖单一算法，而是将多种公认安全的算法集成在一个通用应用程序中。

2.2 PGP 使用的算法体系

PGP 采用混合加密体制 (Hybrid Cryptosystem)，结合了公钥密码体制的便利性和对称密码体制的高效率。

功能	算法 / 标准	说明
数字签名	DSS/SHA 或 RSA/SHA	确保认证性、完整性、不可否认性
消息加密	CAST-128 / IDEA / 3DES	对称加密：用于加密邮件正文（速度快）
密钥交换	RSA 或 Diffie- Hellman	公钥加密：用于加密会话密钥（Session Key）
数据压缩	ZIP	签名之后，加密之前进行
邮件兼容	Radix-64 (Base64)	将二进制密文转换为 ASCII 文本

2.3 符号约定 (Notation)

为了描述严谨，采用以下密码学术语及符号：

- K_S ：一次性会话密钥 (Session Key, 128-bit)
- KR_A/KU_A ：用户 A 的私钥 / 公钥
- KR_B/KU_B ：用户 B 的私钥 / 公钥
- $H(\cdot)$ ：Hash 函数 (如 SHA-1)
- $E_{KU}(\cdot)/D_{KR}(\cdot)$ ：公钥加密 / 私钥解密 (Public-Key Op)
- $E_{KR}(\cdot)/D_{KU}(\cdot)$ ：私钥签名 / 公钥验签 (Public-Key Op)
- $E_K(\cdot)/D_K(\cdot)$ ：对称加密 / 对称解密 (Symmetric Op)
- $Z(\cdot)$ ：ZIP 压缩算法
- $R64(\cdot)$ ：Radix-64 编码
- $||$ ：连接操作 (Concatenation)

3. PGP 完整工作流程 (The PGP Protocol Flow)

本节描述 Alice 向 Bob 发送一封既签名又加密的邮件的完整过程。这是 PGP 最典型的应用场景，涵盖了保密性、认证性、完整性和不可否认性。

3.1 符号定义 (Notation)

为了描述严谨，规定以下符号：

- M ：原始邮件内容 (Message)

- H : Hash 函数 (SHA-1)
- Z : 压缩算法 (ZIP)
- $R64$: Radix-64 (Base64) 转换
- K_S : 一次性会话密钥 (Session Key)
- KR_A, KU_A : Alice 的私钥、公钥
- KR_B, KU_B : Bob 的私钥、公钥
- $E_{Key}(\cdot), D_{Key}(\cdot)$: 加密与解密操作

3.2 发送方流程 (Sender: Alice)

Alice 的 PGP 客户端将执行以下四个步骤：签名 \rightarrow 压缩 \rightarrow 加密 \rightarrow 编码。

3.2.1 Step 1: 数字签名 (Digital Signature)

- 目的：身份认证、消息完整性、不可否认性。
- 操作：
 - 对明文 M 计算哈希值： $h = H(M)$ 。
 - 使用 **Alice** 的私钥 (KR_A) 对 h 进行加密，生成签名 S 。
 - 将签名拼接到明文后面。
- 公式：

$$S = E_{KR_A}(H(M))$$

$$\text{Data}_1 = M \parallel S$$

3.2.2 Step 2: 数据压缩 (Compression)

- 目的：减少数据冗余（增强安全性）、节省传输带宽。
- 注意：压缩必须在签名之后（为了存储方便，签名通常针对明文），加密之前（压缩后的数据熵高，更难被密码分析）。
- 操作：
 - 对“明文+签名”的组合体进行 ZIP 压缩。
- 公式：

$$T = Z(M \parallel S)$$

3.2.3 Step 3: 混合加密 (Hybrid Encryption)

- 目的：保密性 (Confidentiality)。

- 操作：
 - i. 生成密钥：生成一个 128-bit 的随机一次性会话密钥 K_S 。
 - ii. 对称加密 (消息)：使用 K_S 对压缩数据 T 进行对称加密（如 CAST-128, IDEA, 3DES），得到密文 C 。
 - iii. 非对称加密 (密钥)：使用 **Bob** 的公钥 (KU_B) 对 K_S 进行加密，得到加密后的密钥 k 。
- 公式：

$$C = E_{K_S}(T) \quad (\text{消息密文})$$

$$k = E_{KU_B}(K_S) \quad (\text{密钥密文})$$

3.2.4 Step 4: 编码 (Radix-64 Encoding)

- 目的：邮件兼容性（将二进制流转换为 ASCII 文本）。
- 操作：
 - i. 将“加密后的密钥 k ”与“消息密文 C ”拼接。
 - ii. 进行 Radix-64 编码。
- 最终发送的数据：

$$\text{Output} = R64(k \parallel C)$$

3.3 接收方流程 (Receiver: Bob)

Bob 收到邮件后，PGP 客户端执行逆向操作：解码 → 解密 → 解压 → 验签。

3.3.1 Step 1: 解码 (Decoding)

- 将邮件内容的 ASCII 字符还原为二进制数据，分离出“密钥密文 k ”和“消息密文 C ”。

3.3.2 Step 2: 解密 (Decryption)

- 解密会话密钥：Bob 使用自己的私钥 (KR_B) 解密 k ，恢复出会话密钥 K_S 。

$$K_S = D_{KR_B}(k)$$

- 解密消息：Bob 使用 K_S 对密文 C 进行对称解密，得到压缩数据 T 。

$$T = D_{K_S}(C)$$

3.3.3 Step 3: 解压缩 (Decompression)

- 对 T 进行解压，分离出 原始明文 M 和 数字签名 S 。

$$M \parallel S = \text{Unzip}(T)$$

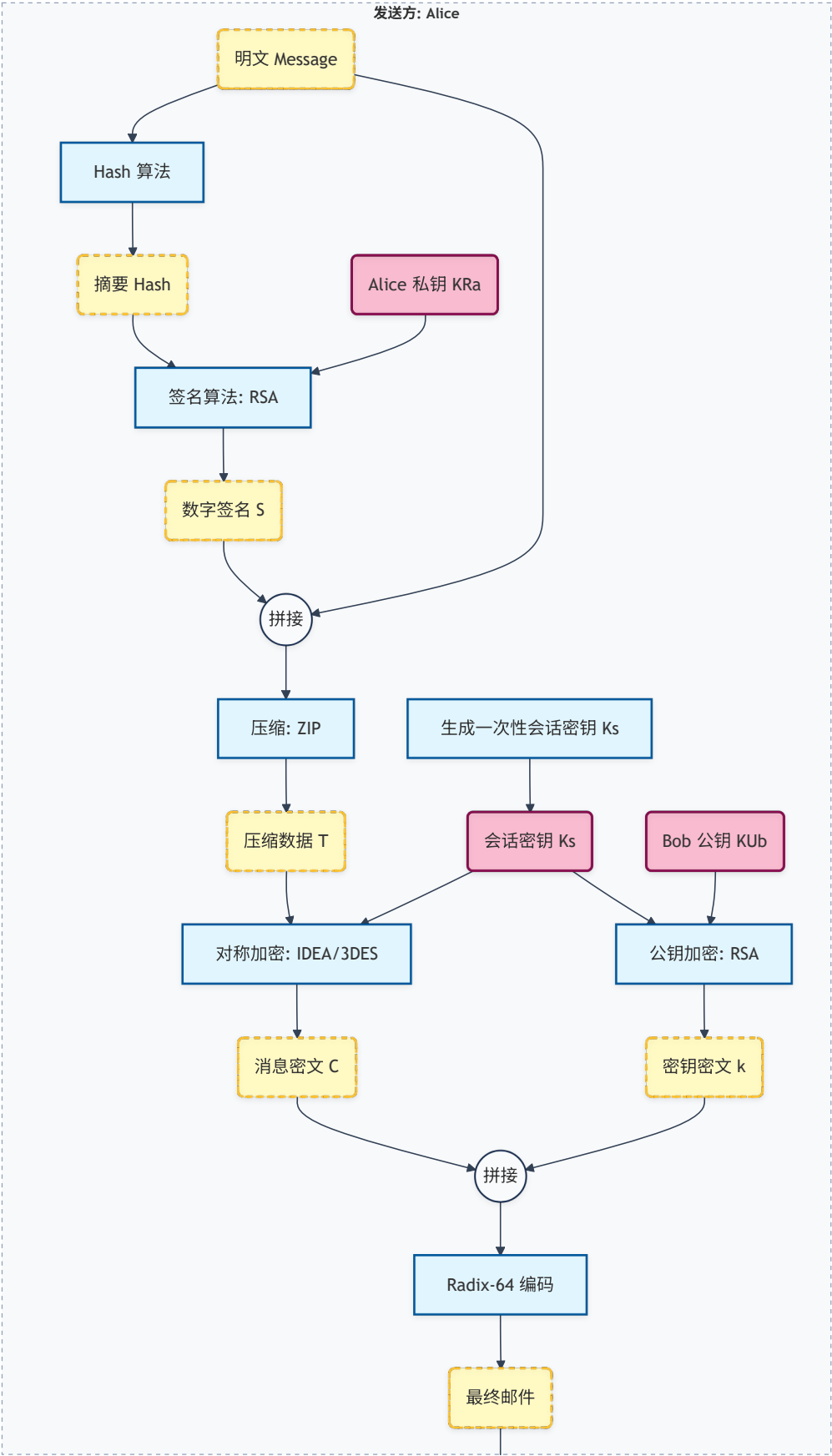
3.3.4 Step 4: 验证签名 (Verification)

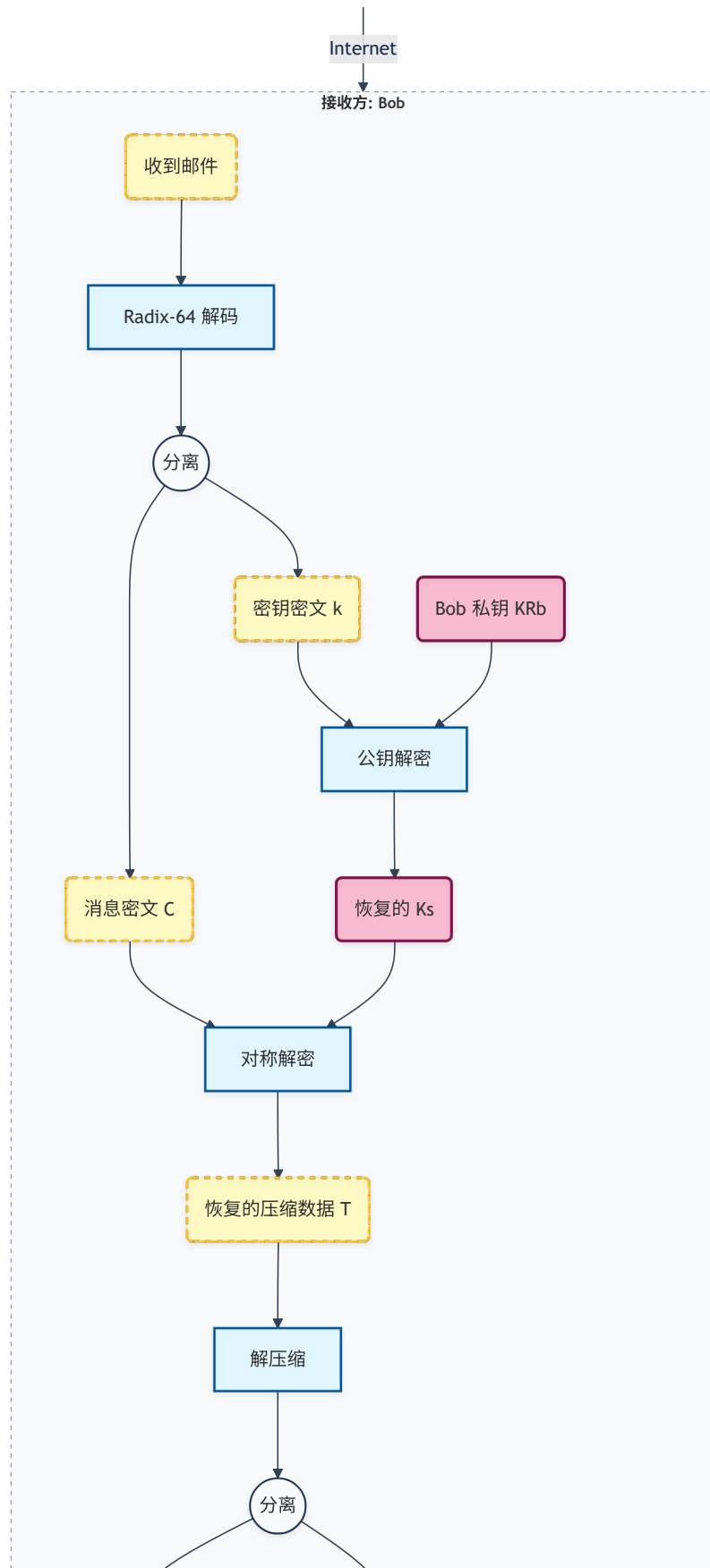
- 计算摘要：Bob 对收到的明文 M 计算哈希值 $h' = H(M)$ 。
- 解密签名：Bob 使用 **Alice** 的公钥 (KU_A) 解密签名 S ，得到原始摘要 h 。

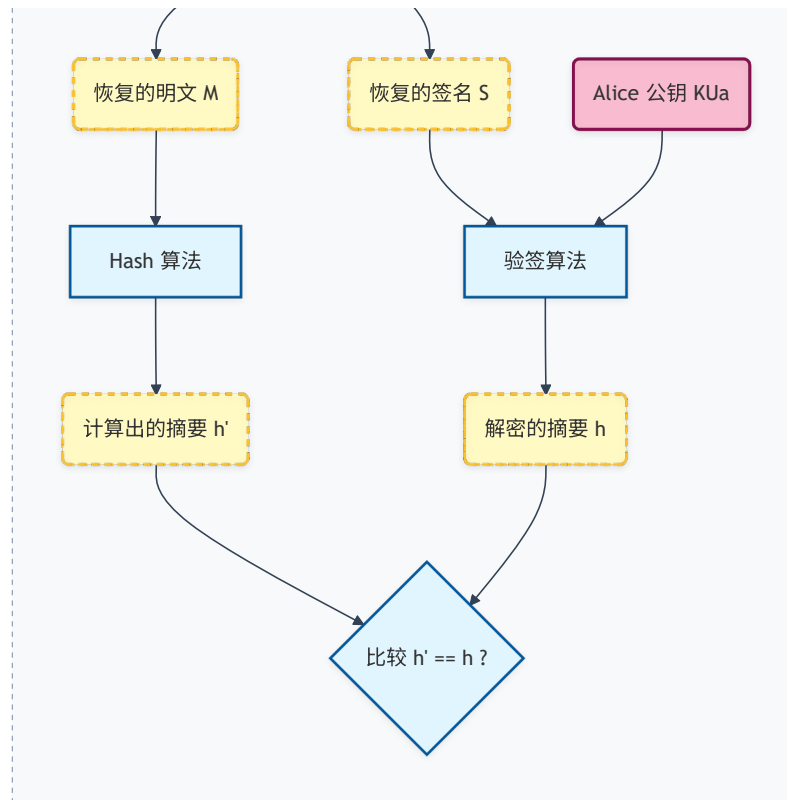
$$h = D_{KU_A}(S)$$

- 比对：判断 $h' == h$ 。
 - 若相等：认证通过（确实是 Alice 发的，且未被篡改）。
 - 若不等：报警，丢弃。

3.4 PGP 完整流程图 (Mermaid Diagram)







3.5 数学模型总结 (Summary Formula)

整个 PGP 发送过程可以用一个嵌套的数学公式概括，解密一层层剥离即可：

$$\text{PGP_Output} = R64 \left[\underbrace{E_{KU_B}(K_S)}_{\text{加密密钥}} \parallel \underbrace{E_{K_S} \left(Z(M \parallel E_{KR_A}(H(M))) \right)}_{\text{加密消息}} \right]$$

- 最内层： $E_{KR_A}(H(M))$ 是对摘要的签名。
- 中间层： $Z(\dots)$ 是压缩。
- 加密层： $E_{K_S}(\dots)$ 是对称加密。
- 最外层： $E_{KU_B}(K_S)$ 保护会话密钥， $R64$ 处理传输格式。

3.6 PGP 密钥管理 (Key Management)

PGP 采用去中心化机制，核心逻辑可概括为：“基于信任网认证的公开发布”。

3.6.1 信任模型：Web of Trust (信任网)

不同于 S/MIME 的层级化 PKI，PGP 建立在用户互信基础上。

- 信任锚点 (**Trust Anchor**): 无中心 CA。每个用户都是独立的“CA”，由用户自己决定信任谁。

- 信任传递 (**Transitivity**):
 - 直接信任: A 验证 B 身份 \rightarrow A 用私钥对 B 公钥签名。
 - 推荐信任 (**Introducer**): A 信任 B, B 信任 C \rightarrow A 间接信任 C (需验证 B 对 C 的签名)。

3.6.2 存储结构: 密钥环 (Key Ring)

客户端通过两个独立的数据结构管理密钥:

1. 私钥环 (**Private Key Ring**):
 - 存储用户自己的公钥/私钥对。
2. 公钥环 (**Public Key Ring**):
 - 存储其他人的公钥。
 - 关键字段: 除了公钥本身, 还包含**“签名列表” (谁认证了这把钥匙)
 - 和“信任级别”** (我多大程度上信赖这个签名者)。

3.6.3 分发与认证机制 (Distribution & Authentication)

本质还是公钥分发的的问题, 这里是使用变相的“公开发布”方法

1. 分发机制: 解决“可访问性” (Accessibility)

- 方式: 公开发布 (**Public Announcement**)。
- 实现: 上传至 **PGP Keyserver**、附在邮件/网页中。
- 风险: 信道不安全, 存在 **MITM** (中间人攻击) 风险。任何人可上传 ID 为 "Bob" 的伪造公钥。

2. 认证机制: 解决“真实性” (Authenticity)

- 核心: PGP 的公钥不仅仅是密钥, 还是 **PGP** 格式的数字证书。
 - 结构: $Certificate = KU_{Bob} || ID_{Bob} || Signatures(by Alice, Carol...)$
- 验证逻辑: 接收方下载公钥后, 不会自动信任, 而是执行以下校验。
 - i. 检查该公钥上是否有签名。
 - ii. 检查签名者 (如 Alice) 是否在我的公钥环中, 且被标记为“受信”。
 - iii. 使用 Alice 的公钥验证签名是否有效。
- 结论: 若签名有效且签名者受信 \rightarrow 接受该公钥。

4. S/MIME (Secure/Multipurpose Internet Mail Extensions)

4.1 简介

S/MIME 是 MIME（多用途互联网邮件扩展）的安全增强版，主要用于工业界和商业环境。它不是像 PGP 那样的单一应用程序，而是内置于邮件客户端（如 Outlook, Thunderbird）的协议标准。

4.2 与 PGP 的核心区别

维度	PGP	S/MIME
应用场景	个人、极客、非正式组织	商业、企业、政府
信任模型	Web of Trust (分布式) 依赖用户间的相互背书	PKI (中心化) 依赖权威 CA 机构 (X.509 证书)
密钥分发	交换公钥文件或上传至 PGP Keyserver	通过 X.509 数字证书传递
主要支持者	开源社区	工业界 (RSA Security, Microsoft 等)

4.3 关键技术组件

- 数字证书：严格遵循 **X.509 v3** 标准。
- **PKI** (公钥基础设施)：包含 CA (证书颁发机构)、RA (注册机构)、CRL (证书吊销列表)。
- 加密流程：与 PGP 逻辑相似（也是混合加密、先签名后加密），但数据格式封装遵循 PKCS 标准。