第 12 章：数字签名

本章要点

- 数字签名是公钥密码学发展过程中最重要的概念之一，它可以提供其他方法难以实现的安全性。
- 数字签名是一种认证机制，它使得消息的产生者可以添加一个起签名作用的码字。通过计算消息的散列值并用产生者的私钥加密散列值来生成签名。
- ElGamal 和 Schnorr 数字签名方案。
- 数字签名标准（DSS）是 NIST 标准。
- 椭圆曲线数字签名（ECDSA）和 RSA 概率签名方案（RSA-PSS）。

主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

数字签名 Digital Signature

- 消息认证可以保护信息交换双方不受第三方攻击，但是它不能处理通信双方自身发生的攻击。
- 例如，当 A 给 B 发送一条认证消息时：
  - 假冒问题：B 可以伪造一条消息并声称该消息发自 A。B 只需要生成一条消息，并用 A、B 共享的密钥产生认证码，并将认证码附于消息之后；
  - 否认问题：A 可以否认曾经给 B 发送过消息。因为 B 可以伪造消息，所以无法证明 A 确实发送过该消息。
- 在收发双方不能完全信任的情况下，需要其他方法来解决这些问题——数字签名。

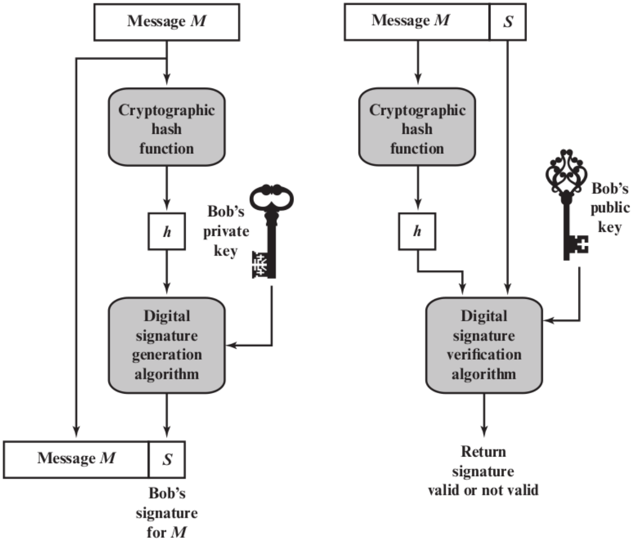
对数字签名的基本要求

- 签名必须是与消息相关的二进制串;
- 签名必须使用发送方某些独有的信息，以防伪造和否认;
- 产生数字签名比较容易;
- 识别和验证数字签名比较容易;
- 伪造数字签名在计算上不可行;
- 保存数字签名的副本是可行的。

数字签名的一般模型

定义

数字签名是使以数字形式存储的明文信息经过特定密码变换生成密文，作为相应明文的签名，使明文信息的接收者能够验证信息确实来自合法用户，以及确认信息发送者身份。



(a) Bob signs a message

(b) Alice verifies the signature

数字签名的基本形式

数字签名的方式：

- 对消息整体的签名：将被签消息整体经过密码变换得到签字；
- 对消息摘要的签名：附在被签消息之后，或嵌在某一特定位置上作一段签字图样。

两类数字签名：

- 确定性数字签名：明文与签名一一对应；
- 概率性数字签名：一个明文可以有多个合法签名，每次都不一样。

直接数字签名

- 直接数字签名指仅涉及通信双方 (发送方和接收方) 的数字签名方案;
- 假定接收方知道发送方的公钥;
- 数字签名通过发送方对整个报文用私钥加密，或只对消息摘要用私钥加密来实现;
- 通常先签名，然后对消息和签名一起加密;
- 安全性依赖于发送方私钥的安全性。

主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

ElGamal 密码体系及数字签名

- 1984 年，T. Elgamal 提出了一种基于离散对数的公开密钥体制，与 Diffie-Hellman 密钥分配体制密切相关。
- ElGamal 密码体制应用于数字签名标准和 S/MIME 电子邮件标准。
- 假定 A 和 B 互相通信，共享大素数 p 和其本原元素 α，满足 $\gcd(\alpha, p) = 1$；
- A 和 B 分别选择私钥 X_A 和 X_B，并计算各自公钥 $Y_A = \alpha^{X_A} \bmod p$ 和 $Y_B = \alpha^{X_B} \bmod p$。

ElGamal 密码体系

加密

- A 选择任意整数 $1 \leq k \leq p-1$；
- 计算 $K = Y_B^k \bmod p$；
- 将 M 加密为密文对 (C_1, C_2)，其中 $C_1 = \alpha^k \bmod p$，$C_2 = KM \bmod p$。

解密

- B 首先恢复 $K = C_1^{X_B} \bmod p = \alpha^{kX_B} \bmod p = Y_B^k \bmod p$；
- 然后恢复明文 $M = C_2 K^{-1} \bmod p$。

ElGamal 数字签名方案

A 签名：用 A 的私钥对消息摘要加密

- 计算 Hash 值 $m = H(M)$，m 满足 $0 \le m \le p-1$；
- 选择任意整数 $1 \le K \le p-1$ 且 $\gcd(K, p-1) = 1$；
- 计算 $S_1 = \alpha^K \bmod p$；
- 计算 $K^{-1} \bmod (p-1)$，即 K 模 $p-1$ 的乘法逆元；
- 计算 $S_2 = K^{-1}(m - X_A S_1) \bmod (p-1)$；
- 得到签名 (S_1, S_2)。

B 验证：用 A 的公钥解密

- 计算 $V_1 = \alpha^m \bmod p$；
- 计算 $V_2 = {Y_A}^{S_1} S_1^{S_2} \bmod p$；
- 如果 $V_1 = V_2$，则签名合法；否则签名不合法。

ElGamal 数字签名方案

$$
\begin{aligned}
V_1 &= V_2 \\
\alpha^m \bmod p &= Y_A^{S_1} S_1^{S_2} \bmod p \\
\alpha^m \bmod p &= \alpha^{X_A S_1} \alpha^{K S_2} \bmod p \\
\alpha^{m - X_A S_1} \bmod p &= \alpha^{K S_2} \bmod p \\
m - X_A S_1 &\equiv K S_2 \bmod (p-1)
\end{aligned}
$$

$$\alpha^i \equiv \alpha^j (\mathrm{mod} p) \iff i \equiv j (\mathrm{mod} (p-1))$$

$$
\begin{aligned}
m - X_A S_1 &\equiv K K^{-1} (m - X_A S_1) \bmod (p-1) \\
m - X_A S_1 &\equiv (m - X_A S_1) \bmod (p-1)
\end{aligned}
$$

ElGamal 数字签名举例

A 产生密钥对：

- 对整数域 $\mathrm{GF}(19)$，即 $p=19$，选择素根 $\alpha=10$；
- A 选择私钥 $X_A=16$，则公钥 $Y_A=\alpha^{X_A} \bmod p=4$。

假设 A 要对 Hash 值为 14 的消息进行签名：

- 选择 $K=5$，满足 $\gcd(K,p-1)=1$；
- $S_1=\alpha^K \bmod p=3$；
- $K^{-1} \bmod (p-1)=11$；
- $S_2=K^{-1}(m-X_AS_1) \bmod (p-1)=4$

B 验证签名：

- $V_1=\alpha^m \bmod p=16$；
- $V_2=Y_A^{S_1}S_1^{S_2} \bmod p=16$；
- 因为 $V_1=V_2$，所以签名合法。

主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

Schnorr 数字签名方案

- Schnorr 签名算法是由德国数学家、密码学家克劳斯・施诺（Claus Schnorr）提出，并于 1990 年申请了专利。该专利于 2008 年 2 月失效，目前该算法可以自由使用。
- Schnorr 签名的特点：计算简便，生成签名的主要工作不依赖于消息，可以在处理器空闲时间完成。
- 选择大素数 p，使得 $p-1$ 包含大素数因子 q；
- 选择整数 α，使得 $\alpha^q \equiv 1(\mathrm{mod} p)$；
- α, p 和 q 公开，作为全局公钥参数。
- 随机选择整数 $0 < s < q$ 作为私钥；
- 计算 $v = \alpha^{-s} \bmod p$ 作为公钥。

Schnorr 数字签名方案

签名

- 选择随机整数 $0 < r < q$，并计算 $x = \alpha^r \bmod p$。注意此步不依赖于消息；
- 将 x 附在消息后面一起计算 Hash 值 $e = H(M\|x)$；
- 计算 $y = (r + se) \bmod q$，得到签名 (e, y)。

验证

- 计算 $x' = \alpha^y v^e \bmod p$；
- 验证是否 $e = H(M\|x')$

$$x' \equiv \alpha^y v^e \equiv \alpha^y \alpha^{-se} \equiv \alpha^{y-se} \equiv \alpha^r \equiv x (\bmod p)$$

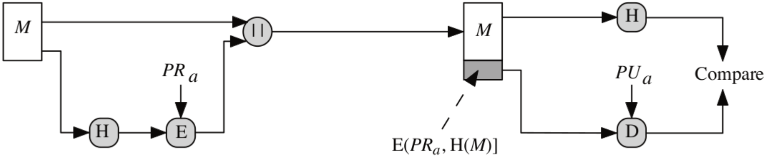
主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

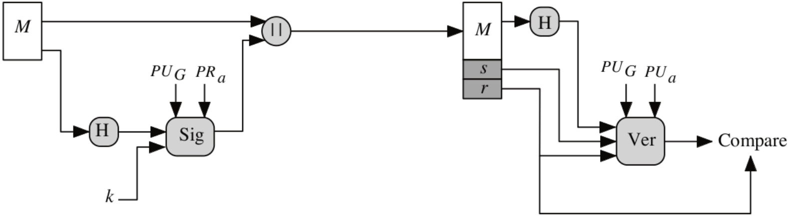
数字签名标准（DSS）

- DSS 是 NIST 作为联邦信息处理标准 FIPS 186 发布的；
- 由 NIST 和 NSA 在 90 年代早期设计；
- DSS 是标准，DSA 是其算法；
- DSS 是 ElGamal 和 Schnorr 算法的变形；
- DSS 使用 SHA 作为 Hash 算法；
- DSS 产生 320 位数字签名，但是具有 512-1024 位的安全性；
- DSS 的安全依赖于 DLP 问题。

DSA vs. RSA



(a) RSA Approach



(b) DSA Approach

与 RSA 相比，DSA 只提供数字签名功能，不能用于加密或密钥交换。

DSA 密钥生成

全局共享参数 p, q, g：

- p：L 位大素数，其中 $512 \leq L \leq 1024$，是 64 整倍数；
- q：$p-1$ 的素因子，长度 N 位（例如 $N = 160$）；
- g：选择 g，使得序列 $\{g^i \bmod p : i = 1, 2, \ldots\}$ 的最小周期为 q，即满足 $g^i \equiv 1 (\mathrm{mod} p)$ 的最小 i 为 q。这里，可以通过计算 $g = h^{(p-1)/q} \bmod p$ 得到 g，其中 $1 < h < p - 1$。

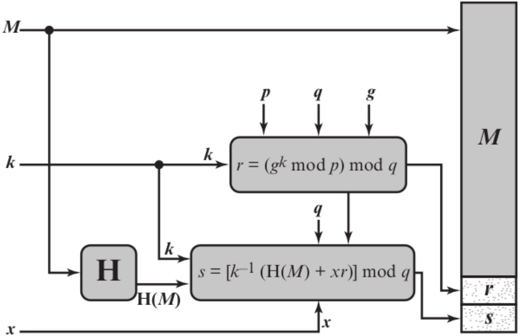
产生用户密钥对：

- 用户选择私钥 $x < q$；
- 用户计算公钥 $y = g^x \bmod p$。

DSA 签名的产生

对消息 M 签名：

1. 计算消息摘要 $m = H(M)$；
2. 产生随机数 $k < q$；
3. 计算 $r = g^k \bmod p \bmod q$；
4. 计算 $s = k^{-1}(m + xr) \bmod q$。

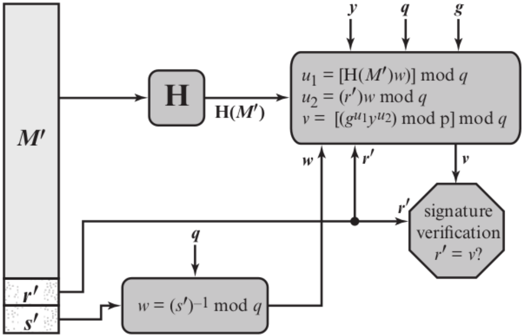


把消息 M 和签名 (r, s) 一起发送给接收方。

DSA 签名的验证

接收方收到消息 M 和签名 (r, s) 后进行验证：

- 计算消息摘要 $m = H(M)$；
- 计算辅助值 $w = s^{-1} \bmod q$；
- 计算辅助值 $u_1 = mw \bmod q$；
- 计算辅助值 $u_2 = rw \bmod q$；
- 计算 $v = g^{u_1} y^{u_2} \bmod p \bmod q$。



如果 $v = r$，则签名合法。

DSA 签名正确性推导

首先，因为 $\{g^i \bmod p\colon i = 1, 2, \ldots\}$ 的最小周期为 q，即 $g^q \equiv 1 \bmod p$，所以 $g^i \equiv g^{i \bmod q} \bmod p$。进而

$$\begin{aligned} v &= g^{u_1} y^{u_2} \bmod p \bmod q \\ &= g^{u_1} g^{x u_2} \bmod p \bmod q \\ &= g^{mw} g^{xrw} \bmod p \bmod q \\ &= g^{mw+xrw} \bmod p \bmod q. \end{aligned}$$

又因为 $s = k^{-1}(m + xr) \bmod q$ 且 $w = s^{-1} \bmod q$，所以

$$k \equiv s^{-1}(m + xr) \equiv w(m + xr) \equiv (mw + xrw) \bmod q.$$

所以

$$r = g^k \bmod p \bmod q = g^{mw+xrw} \bmod p \bmod q.$$

当 $v = r$ 时，可以说明签名是合法的。

DSA

Global Public-Key Components

p prime number where $2^{L-1} < p < 2^L$
for $512 \le L \le 1024$ and L a multiple of 64;
i.e., bit length L between 512 and 1024 bits
in increments of 64 bits

q prime divisor of $(p - 1)$, where $2^{N-1} < q < 2^N$
i.e., bit length of N bits

g $= h(\mathrm{p} - 1)/\mathrm{q}$ is an exponent mod p,
where h is any integer with $1 < h < (p - 1)$
such that $h^{(p-1)/q} \bmod p > 1$

User's Private Key

x random or pseudorandom integer with $0 < x < q$

User's Public Key

$y = g^x \bmod p$

User's Per-Message Secret Number

k random or pseudorandom integer with $0 < k < q$

Signing

$r = (g^k \bmod p) \bmod q$

$s = [k^{-1} (H(M) + xr)] \bmod q$

Signature $= (r, s)$

Verifying

$w = (s')^{-1} \bmod q$

$u_1 = [H(M')w] \bmod q$

$u_2 = (r')w \bmod q$

$v = [(g^{u1} y^{u2}) \bmod p] \bmod q$

TEST: $v = r'$

M = message to be signed

H(M) = hash of M using SHA-1

M', r', s' = received versions of M, r, s

主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

椭圆曲线数字签名（ECDSA）

- 2009 年修订的 FIPS 186 加入了椭圆曲线数字签名算法（ECDSA）；
- ECDSA 密码效率较高，可以使用较短密钥，ECDSA 越来越流行；
- ECDSA 主要包括四个部分：
  - 确定全局参数，包括椭圆曲线参数及基准点；
  - 签名者产生密钥对；
  - 对消息产生 Hash 值，签名者使用私钥、全局参数、Hash 值生成签名；
  - 验证者使用签名者公钥、全局参数验证签名是否合法。

ECDSA 全局参数及密钥产生

- 以 $\mathrm{GF}(p)$ 上的素数域椭圆曲线为例，全局参数包括：
  - 选择随机整数 p 为一个大素数；
  - a, b 为椭圆曲线参数；
  - G 为基准点；
  - n 为点 G 的阶，即满足 $nG = O$ 的最小正整数。
- 每个签名者产生自己的公钥私钥对：
  - 选择随机整数 d 作为私钥，d 满足 $1 \leq d \leq n-1$；
  - 计算公钥 $Q = dG$。

ECDSA 数字签名的产生

为消息 M 产生签名：

1. 计算消息摘要 $m = H(M)$；
2. 选择随机整数 $k \leq n - 1$；
3. 计算 $P = (x, y) = kG$，以及 $r = x \bmod n$。
4. 计算 $s = k^{-1}(m + dr) \bmod n$。

消息 M 的签名为 (r, s)。

ECDSA 数字签名的验证

验证消息 M 的签名 (r, s) 是否合法：

1. 计算消息摘要 $m = H(M)$；
2. 计算辅助值 $w = s^{-1} \bmod n$；
3. 计算辅助值 $u_1 = mw$ 和 $u_2 = rw$；
4. 计算 $X = (x_1, y_1) = u_1G + u_2Q$；
5. 计算 $v = x_1 \bmod n$；

当 $v = r$ 时，接受该签名。

ECDSA 数字签名的验证

因为

$$u_1G + u_2Q = u_1G + u_2dG = (u_1 + u_2d)G = \big((u_1 + u_2d) \bmod n\big)G.$$

又因为 $s = k^{-1}(m + dr) \bmod n$，所以

$$\begin{aligned} k &= s^{-1}(m + dr) \bmod n \\ &= w(m + dr) \bmod n \\ &= (mw + rwd) \bmod n \\ &= (u_1 + u_2d) \bmod n. \end{aligned}$$

因此 $u_1G + u_2Q = kG$。
在验证时，有 $v = x_1 \bmod n$，其中 $(x_1, y_1) = u_1G + u_2Q = KG$；
在签名时，有 $r = x \bmod n$，其中 $(x, y) = KG$。
故当 $v = r$ 时，签名合法。

主要内容

1. 数字签名概述
2. ElGamal 数字签名方案
3. Schnorr 数字签名方案
4. 数字签名标准 DSS
5. 椭圆曲线数字签名 ECDSA
6. RSA-PSS 数字签名算法

RSA-PSS 数字签名算法

- 2009 年修订的 FIPS 186 中，除了 NIST 数字签名算法 DSA 和 ECDSA 之外，还包括几个由 RSA 实验室设计的数字签名方案。
- RSA 概率签名方案（RSA-PSS）被 RSA 实验室推荐为 RSA 各类方案中最安全的签名方案。
- 与其他基于 RSA 的方案不同，RSA-PSS 使用了随机化的处理过程，能够证明其安全性与 RSA 算法的安全性相关。
- 概率数字签名指一个明文可以有多个合法签名，每次签名都不一样。

掩码产生函数（MGF）

- 掩码产生函数 $\mathrm{MGF}(X, \mathrm{maskLen})$：
  - 输入：X 为任意长度位串，maskLen 为输出掩码的字节长度；
  - 输出：长度为 maskLen 字节的串。

- MGF 通常基于安全 Hash 函数来构造。

基于密码学 Hash 函数的 MGF

1. **Initialize variables.**
   ```
   T = empty string
   k = ⌈maskLen/hLen⌉ - 1
   ```
2. **Calculate intermediate values.**
   ```
   for counter = 0 to k
   Represent counter as a 32-bit string C
   T = T ‖ Hash(X ‖ C)
   ```
3. **Output results.**
   ```
   mask = the leading maskLen octets of T
   ```

（注：hLen 为 Hash 函数的输出字节长度）

即 $\mathrm{MGF}(X, \mathrm{maskLen})$ 输出

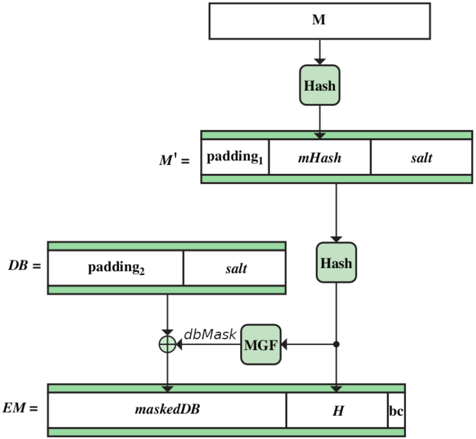
$$\mathrm{Hash}(X||0)||\mathrm{Hash}(X||1)\cdots\mathrm{Hash}(X||k)$$

的前 maskLen 字节。

RSA-PSS 消息编码

由消息 M 生成固定长度的消息摘要，称为消息编码。

1. 生成 M 的 Hash 值 $\mathrm{mHash}=\mathrm{Hash}(M)$；
2. 生成伪随机字节串作为盐，得到数据块 M'；
3. 生成 M' 的 Hash 值：$H=\mathrm{Hash}(M')$；
4. 构造数据块 DB；
5. 计算 H 的 MGF：
   $\mathrm{dbMask}=\mathrm{MGF}(H,\mathrm{emLen}-\mathrm{hLen}-1)$；
6. 计算 $\mathrm{maskedDB}=\mathrm{DB}\oplus\mathrm{dbMask}$；
7. 得到消息编码 $\mathrm{EM}=\mathrm{maskedDB}||H||bc$。

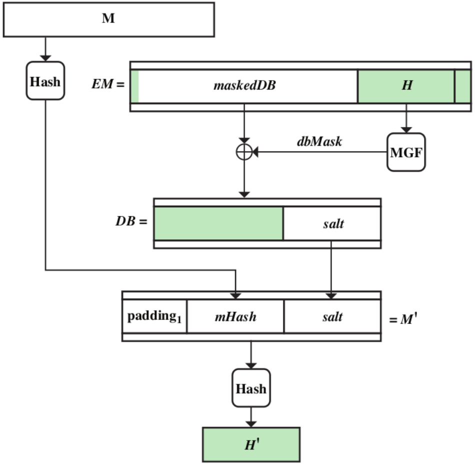


RSA-PSS 签名

- 公钥 e，私钥 d，模数 n；
- 将消息编码 EM 当作无符号非负二进制整数 m；
- 计算签名 $s = m^d \bmod n$；
- 将签名 s 附加在消息 M 后发送给接收方。

RSA-PSS 签名验证

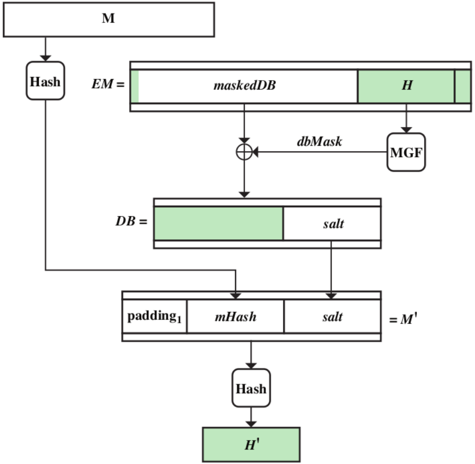
1. 接收方解密 s 得到 $m = s^e \bmod n$，转换为 EM，进而得到 maskedDB 和 H；
2. $\text{dbMask} = \text{MGF}(H, \text{emLen} - \text{hLen} - 1)$；
3. $\text{DB} = \text{dbMask} \oplus \text{maskedDB}$；
4. $\text{mHash} = \text{Hash}(M)$；
5. 构造数据块 M'；
6. $H' = \text{Hash}(M')$；
7. 如果 $H = H'$，则签名合法。



- 盐值使得使用相同私钥对相同消息产生不同签名；
- 验证者不需要知道盐值，不需要去进行盐值比较；
- 盐值的作用类似于 DSA 和 ECDSA 中的伪随机变量 k。

RSA-PSS 签名验证

1. 接收方解密 s 得到 $m = s^e \bmod n$，转换为 EM，进而得到 maskedDB 和 H；
2. $\text{dbMask} = \text{MGF}(H, \text{emLen} - \text{hLen} - 1)$；
3. $\text{DB} = \text{dbMask} \oplus \text{maskedDB}$；
4. $\text{mHash} = \text{Hash}(M)$；
5. 构造数据块 M'；
6. $H' = \text{Hash}(M')$；
7. 如果 $H = H'$，则签名合法。



- 盐值使得使用相同私钥对相同消息产生不同签名；
- 验证者不需要知道盐值，不需要去进行盐值比较；
- 盐值的作用类似于 DSA 和 ECDSA 中的伪随机变量 k。

小结

1. 数字签名概述

2. ElGamal 数字签名方案

3. Schnorr 数字签名方案

4. 数字签名标准 DSS

5. 椭圆曲线数字签名 ECDSA

6. RSA-PSS 数字签名算法