Lec 10. 认证协议

数字签名：给以电子形式传输的消息进行签名，签名后的消息可以通过计算机网络传输。

- 相比于传统文件，数字签名必须签署到文件，即以某种形式将签名绑定到所签的文件上，并且使用公开的验证算法验证，任何人都能验证签名。数据签名文件的copy与原文件签名相同，因此签名文件本身应包含日期等信息。
- 提出目的：避免双方相互欺骗；B 伪造消息/A 声称未发送。
- **特征**
  - 收方能确认或证实发方的签字但不能伪造；
  - 发方发出签名后的消息就不能否认所签消息；
  - 第三者可以确认收发双方之间的消息传送，但不能伪造这一过程。
- **要求**
  - 签名必须是依赖于被签名信息的比特模式；
  - 签名必须使用某些发送者独有的信息，以防止双方的伪造与否认；
  - 签名过程及验证过程容易；
  - 伪造该数字签名在计算复杂性意义上不可行（既包括对一个已有数字签名构造新信息，有包括对一个给定消息伪造一个数字签名）。
  - 在存储器中保存一个数字签名备份是现实可行的。
- **分类**：直接数字签名（仅涉及通信双方，有效性依赖发方密钥的安全性）、仲裁数字签名（使用第三方认证）。

直接数字签名

1. $A \to B: Sig_{SK_A}[M]$ 提供了认证与签名。
   - 只有 A 具有 SK_A，可以用其进行签名；
   - （希望）签名传输中无法被篡改；
   - 需要某些格式/信息冗余度；
   - 任何第三方可以用 PK_A 验证签名。
2. $A \to B: E_K(M||Sig_{SK_A}[H(M)])$ 提供保密的认证与数字签名。
   - 为什么要 $H(M)$？使签名长度固定，更加安全。
   - 只有 A 能够生成 $Sig_{SK_A}[H(M)]$。

实用的 RSA 签名方案

- 密钥的生成（同 RSA 加密系统）
  - 公钥 $P_k = \{e, n\}$，H 是一个安全散列函数；
  - 私钥 $S_k = \{d, n\}$。
- 签名过程：对明文 M，生成签名 $S = H^d(M) \bmod n$
- 验证过程：给定 M, S，如果 $Ver(M, S) = 1$ 当且仅当 $H(M) = S^e \bmod n$。
- **不加散列函数的问题**：
  - 随机产生任何数字都是一个签名。

    - 如果 (M, δ) 是一个已有签名，不加散列函数时 (M^2, δ^2) 也是一个合法签名！
    - 同时，根据 RSA 的乘法同态特性，攻击者可对消息 $M' = Mr$ 查询签名，结果在消去 r 得到 M 的签名。
    - 若攻击者将窃听到的 M^e 发送进行签名，则若进行签名则可以直接得到 M（签名者公私钥对既用作加密又用作签名）！
    - 一个很长的文件，对所有分组都要签名则签名长度很长。
- 添加散列函数后，可以证明数字签名算法是安全的（如果攻击者可伪造签名，则可以解决 RSA 问题或攻破哈希函数）。

PKCS1 签名（长哈希）：是使用最广泛的数字签名。格式如下：

```
D: 00 01 | FF FF FF ... FF FF 00 DI | H(m)
```

其中，H 为抗碰撞 Hash，长度为 h，要求 $h < t - 88$ （t 为签名总长度）；`DI` 为对哈希函数 H 的编码；`FF` 为填充，使得加上 $H(m)$ 的长度正好是 t 比特；用整个 D 用 RSA 签名。

- 为何要填充至固定长度？避免 $H(m_1) = p_1, H(m_2) = p_2, H(m_3) = p_1 p_2$ （其中 $m_3 = m_1 m_2$）的选择明文攻击，注意到这里利用到了 RSA 的乘法同态特性。
- 阅读方式最好从右往左，保证 $H(m)$ 后没有其他信息！（从前往后进行验证会出现错误，原因可能是没有验证 $H(m)$ 后是否还有其他信息）。否则有如下攻击方式：

  选择 M' 得到相应的 $H(M')$，进行适当填充使得目前的 D' 长度小于 $\frac{t}{3}$；并在最后补 0 至长度为 t 后得到 D，开 3 次方根得到签名 S；假如 RSA 签名的公钥 $e = 3$，那么验证时 $S^e = S^3 = T$，从前往后检验 T 将会验证成功，而实际上验证失败。

数字签名标准 DSS：利用了 Hash 函数和 DSA 算法进行数字签名。

- **初始化**：全局公钥 (p, q, g)，p 为 1024 比特以上大素数，q 是 $(p-1)$ 的素因子，为 160 比特的素数，$g = h^{(p-1)/q}$，且 $1 < h < (p-1)$ 使得 $h^{(p-1)/q} \bmod p > 1$，g 的阶为 q。
- **签名**：用户对每个需要签名的消息选择秘密随机数 k 满足 $0 < k < q$；则对报文 M，签名为 (r, s)
    - $r \equiv g^k \pmod p \pmod q$
    - $s \equiv [k^{-1}(H(M) + xr)] \pmod q$
- **验证**：验证者对于收到的 (M, r, s) 计算
    - $w \equiv s^{-1} \pmod q$
    - $a \equiv H(M)w \pmod q$
    - $b = rw \pmod q$
    - $v \equiv g^a y^b \pmod p \pmod q$
    - 当且仅当 $v = r$，验证成功。
- 如果 k 暴露了，则攻击者可以从签名中恢复出签名私钥。如果对两个不同的消息使用了同样的 k 进行签名，攻击者也可以恢复出签名私钥（对 Sony PS3 的攻击）。

ECC 签名算法 ECDSA：利用了 ECC 和 DSA 算法进行签名。

- **初始化**：椭圆曲线 $E(GF(q))$，基点 G，阶为素数 n，$h = |E|$。公钥 $Q \in E(F_q)$，私钥 $d \in [1, n-1]$ 使得 $Q = dG$。
- **签名**：随机选取 $k \in [1, n-1]$，计算 $kG = (x_1, y_1)$ 与 $r = x_1 \bmod n$（如果 $r = 0$ 则重新选择 k）；然后计算 $s = k^{-1}[H(m) + dr] \bmod n$（如果 $s = 0$ 则重新选择 k），则 m 的签名是

(r, s)。

- **解密**：B 收到 $(m, (r, s))$ 验证签名，计算 $e = H(m)$，$u_1 = es^{-1} \bmod n$ 且 $u_2 = rs^{-1} \bmod n$；计算 $(x_1, y_1) = u_1 G + u_2 Q$，如果 $x_1 = r \bmod n$，则签名正确。
- 注意到可以通过只发 x 来减小签名长度，不过会产生两个合法签名，一般情况下并不会产生危险；但是在比特币交易所交易时可能会产生漏洞，造成交易所损失（可以硬编码纵坐标小于 $p/2$ 来解决问题）

其他类型的签名

- **盲签名**：签名着不知道所签署文件的内容；核心：不可追踪。
- **群签名**：一组人拥有签名的权利，每个人的签名等效（群组中成员无法爆料消息）；
  - **环签名**：群签名的变体，可以通过本人私钥+全体其他群成员公钥加密消息进行爆料，并通过全体成员公钥进行验证。
- **门限签名**：一组人中只有达到规定数目人的同意，才能产生签名；
- **代理签名**：签名人临时将签名券交由代理人行使；
- **属性签名**：签名的权限随身份改变而调整；
- **多重签名**：多个人的会签；
- Lamport 数字签名：数字签名完全可以不需要“陷门单向函数”，仅使用“单向函数”即可。
- ……

公钥加密和签名算法的两大问题：密钥如何管理？（难点：不能轻易接受其他人的公钥；特点：关键不在于加密，而是认证）；Hash 函数的安全性？（不安全的哈希函数直接导致签名不安全）

公钥的分配途径：

- **公钥的公开发布**：任一通信方将公钥发送给另一方或广播给通信各方，如 PGP 中用户在给公开论坛发送消息时，将其公钥附加在钥发送的消息后。问题在于：任何人都可以伪造公钥的发布过程，知道公钥发现者发现假冒。
- **公开可访问目录**：通过在一个公开目录中注册密钥来获得更大的安全性；公开目录必须可信，且具有如下性质：
  - 包含目录项 {姓名，公钥}；
  - 通信成员必须经过安全注册（亲自或安全新到）；
  - 通信成员任何时刻均可更新公钥；
  - 目录定期更新；
  - 通信方可以访问电子目录。

  缺点：
  - 一旦攻击者获得目录管理员私钥，则可伪造任意通信方；
  - 攻击者还可能修改目录管理员保存的记录来达到目标。
- **公钥授权**
  - A 发送代带时间戳的消息给公钥管理员，请求 B 的当前公钥；
  - 管理员用自己的私钥签名一条消息（含 B 的公钥、A 的请求、原始时间戳）给 A，A 可亲自确认消息来自管理员；
  - A 保存 B 的公钥，生成一个临时“挑战” [ID(A) || N1] 通知 B；
  - B 用第 1～2 步获取 A 的公钥；并对 A 的挑战响应，并传输一个新的“挑战” [N1 || N2]；
  - A 响应 B 的挑战 [N2]。

问题：

- 只能从管理机构获得公钥；公钥管理员称为瓶颈；
- 容易产生 DOS 攻击；
- 管理机构如何得到 A 的公钥？A 如何得到管理机构的公钥?
- 公钥管理员维护的 {姓名，公钥} 目录容易被篡改。

- **公钥证书**：由 Kohnfelder 提出，通信各方使用证书来交换密钥，而不是通过公钥管理员；安全性与从公钥管理员出获得密钥的可靠性相同。证书由证书管理员产生，包含公钥以及其他信息，发给拥有相应私钥的通信方，通信一方通过传递证书将公钥传给另一方，其他人可验证证书确实由管理员产生。这是一种更加现实的方法。
  - **要求**：
    - 任何人都可读取证书并确认证书拥有者的姓名和公钥；
    - 任何人都可验证证书是真实的；
    - 任何人都可验证证书是新鲜的（加入时间戳，时间一到更新公钥）；
    - 只有证书管理员才能产生和更新证书。
  - **效果**：类似信用卡。
  - **具体过程**：A 与 B 分别向公开密钥管理机构报告各自公钥，公开密钥管理机构签发机构的数字签名证书证明公钥的真实、新鲜性，然后通信双方互相进行证书交换即可建立连接。
  - **问题**：A 如何得到证书管理员（CA）的公钥？密钥丢失后如何挂失？（证书撤销列表）

利用公钥密码分配会话密钥

- **简单的秘密分配**：A 生成公私钥对 PK_A, SK_A，并将 PK_A, ID_A 发送给 B；B 用 A 的公钥加密会话密钥 K_S 返回给 A；然后 A 放弃公私钥对，B 放弃 A 的公钥，即

```
(1) A send (PKA, IDA) to B;
(2) B response E(KS, PKA) to A;
(3) A and B get session key Ks, and drop their PK/SK.
```

  插入攻击：攻击者 Malice 截获 A 发送给 B 的消息并篡改为 PK_M, ID_A；当 B 返回用 M 的公钥加密的会话密钥时，攻击者 Malice 解密并再次用 A 的公钥加密返回给 A；则 A 与 B 都知道 K_S，但是他们不知道攻击者 Malice 也知道 K_S。

- **具有保密性和真实性的密钥分配**：
  - A 用 B 的公钥对自己的身份 ID_A 和临时交互号 N_1 加密发送给 B；
  - B 用 A 公钥回复消息 N_1, N_2 （N_2 为了确认 A 身份）；
  - A 用 B 公钥回复信息 N_2（已经确认 B 身份，B 收到后也能确认 A 身份）；
  - A 发送会话密钥并发送带数字签名的会话密钥给 B，供 B 验证是 A 发送的；
  - A 和 B 都能获得会话密钥 K_S。

```
(1) A send E(N1 || ID(A), PKB) to B;
(2) B response E(N1 || N2, PKA) to A;
(3) A response E(N2, PKB) to B;
(4) A choose session key Ks, and send E(Ks || Sig(Ks, SKA), PKB) to B;
(5) A and B get session key Ks.
```

双方密钥交换：Diffie-Hellman 密钥交换协议。