密码学第十次作业

1.数字签名的基础知识

(1) 对数字消息进行签名，以防消息的冒名伪造或篡改，也可用于通信双方的身份鉴别。

(2) M：一个可能消息的有限集　A：一个可能签名的有限集　K：一个可能密钥的有限集　S：一个可能的签名算法集合　V：一个可能的验证算法的集合

(3) Gen：密钥生成算法　输出一对密钥(pk, sk)

Sign：签名算法　一条私钥sk

Vrfy：验证算法　公钥pk和签名σ

(4) 唯密钥攻击　已知消息攻击　一般选择消息攻击　定向选择消息攻击　适应性选择攻击

完全破译　通用伪造　选择伪造　存在性伪造

(5) 鉴别　密钥

1. 验证身份：数字签名应能够验证签名者的身份，确保签名是由合法的签名者生成的。
2. 数据完整性：数字签名应能够验证数据的完整性，即在传输或存储过程中，数据没有被篡改或损坏。
3. 不可抵赖性：数字签名应能够防止签名者否认其签名的事实。即签名者不能以后否认签署过该文件或数据。
4. 不可篡改性：数字签名应具备抵抗篡改的能力，一旦签名生成，任何人都不能对签名内容进行更改。
5. 时间戳：数字签名可以附加一个时间戳，以证明签名是在特定时间之前或之后创建的。
6. 快速生成：数字签名的生成过程应该高效、快速，以便在实时交互或大规模应用中使用。
7. 安全性：数字签名应具备高度的安全性，防止未经授权的个人或恶意方对签名进行伪造或破解。
8. 可扩展性：数字签名方案应该具备可扩展性，能够适应不同规模和类型的应用场景。
9. 证书认证：数字签名通常会与证书机构的认证过程相关联，以验证签名者的身份和信任级别。

2.数字签名的具体实现方案

(1) $I.$ 公钥和私钥　离散对数关系　随机整数x　使得$1 < x < p-1$　计算$y = g^x \bmod p$　x　(y, p, q)

$II.\ K < q$

- $y = \alpha^{X_A} \bmod p = 10$
- $S_1 = \alpha^k \bmod p = 2$
- $S_2 = k^{-1}(m - X_A S_1) \bmod (p-1) = 4$　签名为：$Sig(X_A, m) = (S_1, S_2) = (2, 4)$
- 验证：
- - $V_1 = \alpha^m \bmod p = 4$
  - $V_2 = y^{S_1} S_1^{S_2} \bmod p = 4$　$V_1 = V_2$ 验证完毕

$III.a.\ xS_1 = (m - S_2 k) \bmod (p-1) = 14 \bmod 18$　$x = 7 \bmod 9$

$b.$ 构造如下方程：

$$m_1 = r_1 \cdot x + s_1 \cdot k = 4 = 2 * x + 4 * k$$

$$m_2 = r_2 \cdot x + s_2 \cdot k = 17 = 2 * x + 15 * k$$

解得：

$$\begin{cases} X_A = 7 \\ K = 11 \end{cases}$$

(2) $I.$ 13

$II. Q = dG = (8, 3)$

$III. P = kG = (x, y) = (2, 7) \quad r = x \bmod n = 2 \quad e = H(m) = 20220529$
$s = k^{-1}(e + dr) \bmod n = 11 \quad$ 签名：$(r, s) = (2, 11)$

$IV. w = s^{-1} \bmod n = 6 \quad u_1 = ew = 121323174 \quad u_2 = rw = 12$
$X = (x_1, y_1) = u_1 G + u_2 Q = (2, 7) \quad v = x_1 \bmod n = 2 \quad$ 可验证$v = r = 2$

(3)

1. 攻击者计算伪造的签名s'，即 $s' = Y^{m^{-1} \bmod (q-1)} \bmod q$。
2. 攻击者将伪造的消息m和签名s'作为合法的签名对进行伪造。

验证伪造签名的过程如下：

1. 接收者收到消息m和签名对(m, s')。
2. 接收者通过计算$s'^m \bmod q$来验证签名的有效性。

   计算 $s'^m \bmod q$ 的结果应该与原始的公钥Y相等。

3.密钥管理与分发

(1)

1. A选择一个密钥后以物理的方式传递给 B
2. 第三方选择密钥后物理地传递给A和B。
3. 如果 A 和 B 先前或者最近使用过一个密钥，则一方可以将新密用旧密钥加密后发送给另一方。
4. 如果A和 B 到第三方C有加密连接，C 可以在加密连接上传送密钥给A和B。

(2) $I.$

1. $A \to S : A, B, N_A$
2. $A \to B : \{N_A, K_{AB}, B, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$
3. $A \to B : \{K_{AB}, A\}_{K_{BS}}$
4. $B \to A : \{N_B\}_{K_{AB}}$
5. $A \to B : \{N_B - 1\}_{K_{AB}}$
6. Alice向服务器发送一条包含她本人和Bob标识的消息，告诉服务器她想和Bob通信。

$$A \to S : A, B, N_A$$

2. 该服务器产生KAB,并发送回Alice一个副本和一个被KBS加密的副本由Alice转交给Bob。由于Alice可能同时发出多份通信验证请求，所有nonce保证响应消息是新的和并与某一请求对应。在响应中加入了Bob的标识以告诉Alice她将与谁共享该密钥。

$$A \to B : \{N_A, K_{AB}, B, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$

3. Alice将KAB密钥转交给Bob，他能通过KBS密钥（他于服务器的共享密钥）解密出该密钥，以验证数据的可靠性。

$$A \to B : \{K_{AB}, A\}_{K_{BS}}$$

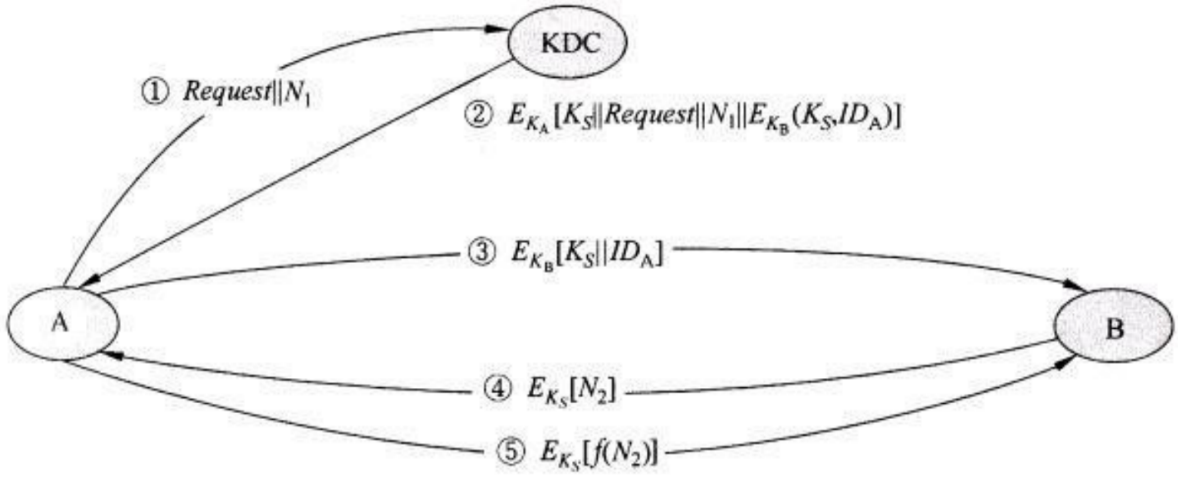
4. 然后Bob想Alice发送一个通过密钥KAB随机数nonce，表示他以获得密钥

$$B \rightarrow A : \{N_B\}_{K_{AB}}$$

5. Alice对接收到的随机数nonce进行简单的操作，重新进行加密，并把它发送回确认她也持有密钥并且仍处于活跃状态。

$$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$$

流程图：



$II.$

1. A向S请求B的公钥

$$A \rightarrow S : A, B$$

2. S响应B的标识和公钥，并使用自己的私钥加密数据，以便A验证自己。

$$S \rightarrow A : \{K_{PB}, B\}_{K_{SS}}$$

3. .A引入随机数NA,然后发送给B

$$A \rightarrow B : \{N_A, A\}_{K_{PB}}$$

4. B向S请求A的公钥

$$B \rightarrow S : B, A$$

5. S的响应

$$S \rightarrow B : \{K_{PA}, A\}_{K_{SS}}$$

6. B引入随机数NB和NA使用KPA加密后发送给A，以证明他的能力。

$$B \rightarrow A : \{N_A, N_B\}_{K_{PA}}$$

7. A发回确认请求包含NB，以证明他能使用KSA解密

$$A \rightarrow B : \{N_B\}_{K_{PB}}$$

(3)

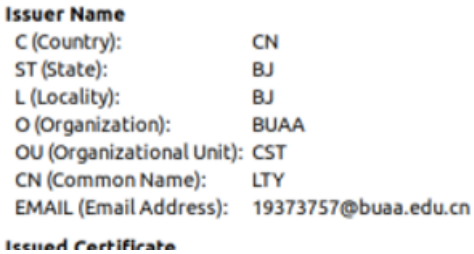
公钥基础设施：（Public Key Infrastructure，PKI）管理和维护公钥密码学体系中的公钥和数字证书的框架和体系结构，它提供了一套规范和流程，用于生成、分发、存储、验证和撤销数字证书，以及进行公钥加密、数字签名和身份验证等安全功能。

目的：

1. 身份认证：PKI提供了一种机制，通过数字证书来验证和证明通信参与者的身份。数字证书是由可信的证书颁发机构（Certificate Authority，CA）签发的，包含了公钥和相关的身份信息，以证明证书持有者的身份。
2. 数据完整性和防篡改：通过使用公钥加密和数字签名技术，PKI可以确保数据在传输过程中的完整性，并能够检测任何未经授权的更改或篡改。接收者可以使用发送者的公钥验证数字签名，确保数据的完整性和真实性。
3. 密钥管理和分发：PKI提供了一种机制来生成、存储和分发公钥和私钥对。它确保了公钥的安全分发，并提供了密钥的保护和管理机制，包括密钥的生成、存储和撤销。
4. 数字证书撤销：PKI允许证书颁发机构在需要时撤销数字证书，例如当私钥丢失、泄露或证书持有者不再可信时。这确保了对已撤销证书的有效性进行验证，并保护通信的安全性。

(4)

第一个证书私钥：CN-BJ-BJ-BUAA-CST-LTY



第二个证书：也是LTY

1.浏览器向服务器发送连接请求，并请求其证书。

2.服务器将其证书和一些其他信息发送回浏览器。

3.浏览器使用内置的信任根证书颁发机构列表，验证服务器证书的合法性。验证包括以下几个方面：

- 证书是否被颁发给正确的域名（即与用户访问的网站域名匹配）。
- 证书是否由可信的颁发机构颁发，是否在浏览器内置的信任根证书颁发机构列表中。
- 证书是否已过期或被吊销。

4.如果证书验证通过，浏览器生成一个随机的对称加密密钥，并使用服务器的公钥进行加密，然后将其发送给服务器。

5.服务器使用其私钥解密浏览器发送的密钥，并使用该密钥加密所有后续的通信数据。

6.浏览器和服务器之间使用对称加密密钥进行通信，从而保证通信数据的机密性和完整性。

如果证书验证不通过，浏览器会发出安全警告，并提示用户是否继续访问该网站。在这种情况下，用户应该仔细考虑是否继续访问该网站，因为它可能存在安全风险。