**信息安全实验221班 2022132006 杨佳伲**

第4次课后作业参考答案散列算法及数字签名

题目1:

(1) Hash 函数的安全性要求有哪些?请具体说明。

(2) Hash函数的用途有哪些?请具体列举至少5种进行说明。

答：(1)

## 抗碰撞性（Collision Resistance）：抗碰撞性指的是在合理的时间内很难找到两个不同的输入x和y使得它们的哈希值相同，即H(x)=H(y)。对于所有x/neqy，找到H(x)=H(y)是不可行的。

## 假设有一个[哈希函数](https://so.csdn.net/so/search?q=%E5%93%88%E5%B8%8C%E5%87%BD%E6%95%B0&spm=1001.2101.3001.7020" \t "_blank) H，其输出是一个128位的哈希值。为了证明这个函数具有抗碰撞性，我们需要展示即使在大量尝试之后也很难找到两个不同的输入导致相同的哈希值。

在密码学中，这通常通过展示哈希函数能够抵抗“生日攻击”来完成。生日攻击是一种概率攻击，其原理类似于生日悖论：在一个23人的房间中，至少有两人生日相同的概率超过50%。

在哈希函数的上下文中，这意味着在 IMG_259 次尝试后，找到一对碰撞的概率超过50%，其中 n 是哈希值的位数。因此，如果我们的哈希函数 H 在超过 IMG_260 次尝试后仍然没有碰撞，我们可以认为它具有良好的抗碰撞性。在实际应用中，如密码存储或数字签名，开发者会选择已知具有强抗碰撞性的哈希函数。

1. 抗第一原象性（Pre-image Resistance）(单向性)：抗第一原象性意味着对于给定的哈希值 h，在合理的时间内很难找到任何原始输入 x，使得 H(x)=h。对于给定的h,找到任意x使得H(x)=h是计算不可行的。这是一个理论上的属性。例如，如果攻击者获得了数据库中的哈希密码，他们不应该能够找到对应的密码。

假设一个黑客获得了一个哈希值 h，他们想要找到一个消息 m 使得 H(m)=h。如果哈希函数是抗第一原象的，黑客尝试所有可能的输入直到他们找到一个匹配的原象，这个过程应该是不切实际的。例如，如果一个哈希函数的输出是256位长，黑客将需要尝试IMG_264个不同的输入才能找到正确的原象，这在现实中是不可行的。

## 抗第二原象性（Second Pre-image Resistance）（抗弱碰撞）：抗第二原象性指的是对于一个固定的输入 x，在合理的时间内很难找到一个不同的输入 y，使得 H(x)=H(y)。对于固定的x和任意的y/neqx,找到H(x)=H(y)是计算不可行的。这保证了即使攻击者知道一个特定的输入和它的哈希值，他们也不能找到另一个具有相同哈希值的不同输入。

## 考虑一个电子文件签名的场景，Alice 用哈希函数 H 对文件 f 生成一个哈希值 h=H(f) 并用她的私钥加密它，创建一个数字签名。Bob 可以用Alice的公钥解密并得到 h，然后自己对文件 f 运行 H 来验证签名。

如果一个攻击者想要伪造一个签名，他们需要找到一个不同的文件 f′ 使得 H(f′)=h。如果哈希函数是抗第二原象的，那么找到这样的 f′ 将是非常困难的。在这些例子中，一个关键的假设是哈希函数的输出空间足够大，这样概率才会站在我们这一边使得找到满足上述条件的输入变得不现实。在实践中，这就是为什么推荐使用比如SHA-256或更高位数哈希函数的原因，因为它们为找到原象或碰撞提供了足够小的概率。

(2)

应用：

1. 数据完整性验证: HASH函数有类似数据冗余校验类似的功能，但是它比简单的冗余校验碰撞的概率要小得多，顾而在现在密码学中总是用HASH来做关键数据的验证。
2. 口令，密码等安全数据的安全存储(单向性的运用）:利用HASH函数的这个特点，我们能够实现口令，密码等安全数据的安全存储。密码等很多关键数据我们需要在数据库中存储，但是在实际运用的过程中，只是作比较操作，顾而我们可以比较HASH结果。这一点相信在银行等系统中有所运用。
3. 数字签名（碰撞约束以及有限固定摘要长度）：数字签名正是运用了这些特点来提高效率的。我们知道非对称加密算法速度较低，通过HASH处理我们可以使其仅仅作用于HASH摘要上，从而提高效率。
4. 可以运用HASH到随机数的生成和密码，salt值等的衍生中：因为HASH算法能够最大限度的保证其唯一性，故而可以运用到关键数据的衍生中（从一个随机的种子数产生，并且不暴露种子本身秘密）。
5. 防止篡改的时间戳：Hash函数还可以用于生成防篡改的时间戳。将文档的哈希值与时间信息一起存储，可以证明数据在特定时间点就存在，并且自那时以来未被更改。

除此之外还可链接数据结构中的应用、实现数据去重和快速查找

题目2:

请具体说明:为何直接对消息进行签名可能会被敌对方伪造签名?为了防止

签名被伪造，应采用什么方式?请分析说明原因。

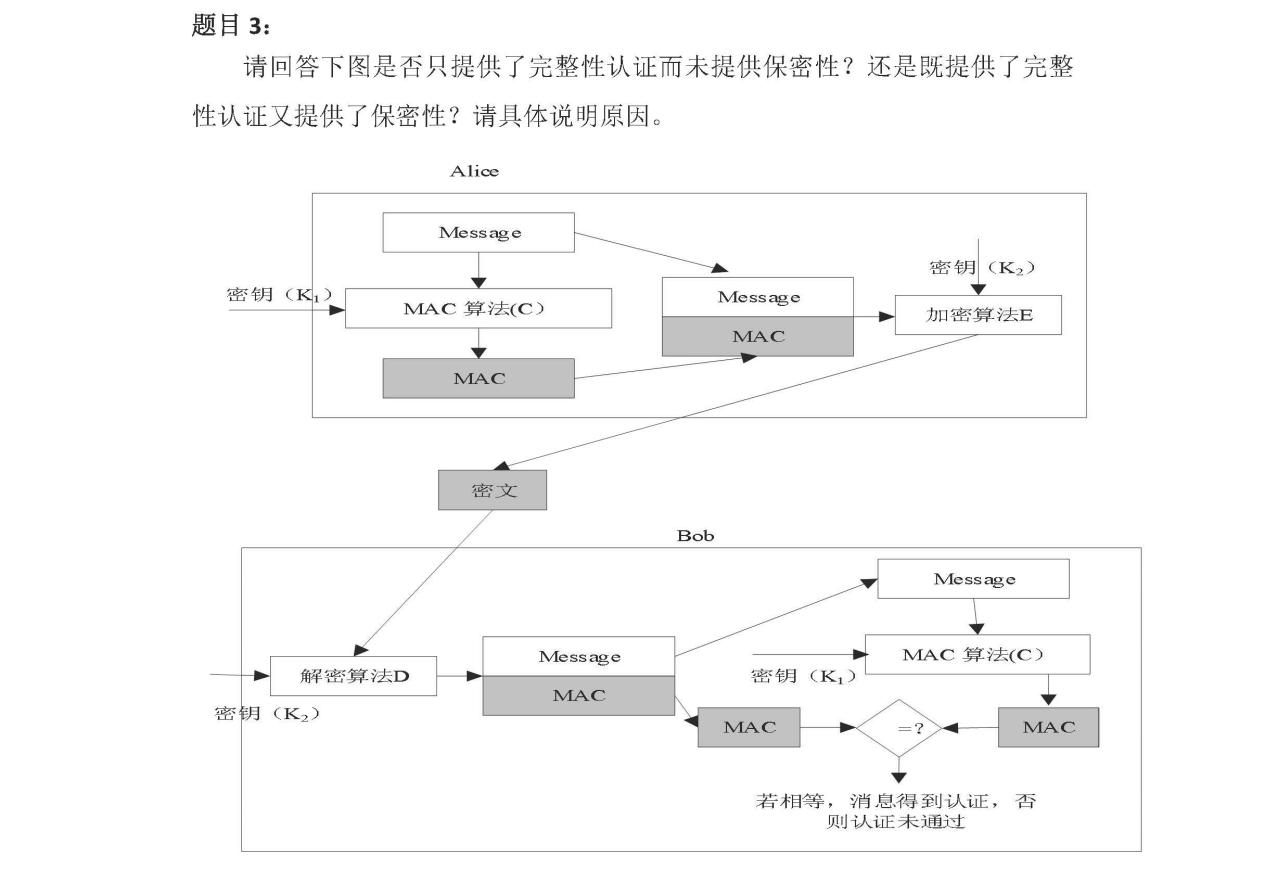
答：直接对消息进行签名可能会被敌对方伪造签名的原因在于，签名过程中未包含足够的安全措施来确保签名的真实性和完整性。如果签名仅仅是对消息直接进行签名，而没有使用任何额外的安全机制，那么攻击者可能通过消息篡改、重放攻击等方式导致伪造签名

为了防止签名被伪造，通常采用的方式是使用数字签名算法，并结合公钥基础设施（PKI）来实现签名的验证和身份认证。具体来说，可以采用以下方式：

1. 使用数字证书：数字证书是由可信任的第三方颁发机构（CA）签发的，用于证明实体身份的数字文档。通过数字证书，可以确保签名的真实性和完整性，因为只有合法的实体才能获得有效的数字证书。
2. 使用公钥基础设施（PKI）：PKI 是一套公钥加密体系，用于建立和管理数字证书、公钥和私钥。通过 PKI，签名验证方可以使用签名者的公钥来验证签名的有效性，从而防止伪造签名。
3. 使用安全哈希算法：在签名过程中，通常会将消息进行哈希处理，然后对哈希值进行签名。这样做可以防止消息的篡改，因为即使消息被篡改，其哈希值也会发生变化，导致签名验证失败。

题目3:

请回答下图是否只提供了完整性认证而未提供保密性?还是既提供了完整性认证又提供了保密性?请具体说明原因。



如图展示了Alice和Bob之间的通信流程。

在Alice端，首先有一个明文消息，然后通过一个MAC算法进行消息认证（标记为C），得到一个密文。接着，Alice使用密钥K2对密文进行加密，得到最终的密文，在这一过程中，MAC和Message一同被加密，实现了消息的保密性。在Bob端，他首先接收到了密文，然后使用密钥K2进行解密，同样地，他也使用MAC算法对收到的密文进行认证。最后，Bob比较两个MAC值是否相等。如果相等，那么消息被认证为正确；如果不等，则认证失败，实现了消息完整性认证。

因此：既提供了完整性认证又提供了保密性。