密码学第九次作业

1.

(1)： 信息安全性　　512bit　　160bit

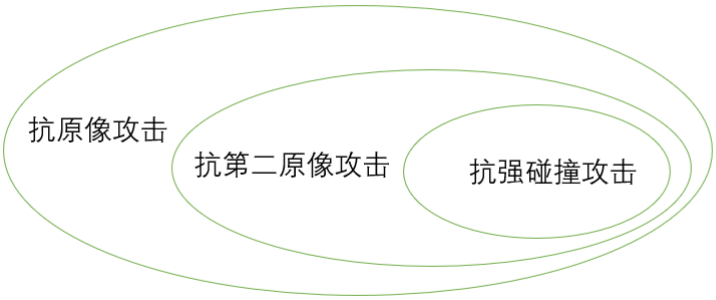
(2)： 消息认证　　数字签名　　伪随机数生成器　　原因：由于现代密码学使用的都是公钥密码技术,而且这种非对称算法的运算速度较慢, 所以对消息在传输前都要进行一定的压缩计算。

(3)： I. 抗原像攻击　单向性　对于任意给定的Hash码h找到满足$H(y)=h$的y在计算上是不可行的

II. 抗第二原像攻击　抗弱碰撞性　对任意给定的分块x，找到满足$y\neq x$且$H(y)=H(x)$的y在计算上是不可行的

III. 抗强碰撞攻击　找到任何满足$H(y)=H(x)$的偶对(x,y) 在计算上是不可行的

关系图:



(4)： 2^m　　$2^{m/2}$

1. 攻击者首先将原始消息 M_1，计算其哈希值 $H(M_1)$。
2. 然后攻击者随机选择一个消息 M_2，计算其哈希值 $H(M_2)$。
3. 如果 $H(M_1)=H(M_2)$，则攻击者找到了一对碰撞。攻击者可以将 M_1 发送给签名方，然后将 M_2 作为伪造的消息。
4. 如果 $H(M_1)\neq H(M_2)$，则攻击者需要继续随机选择消息，重复上述过程，直到找到一对碰撞为止。
5. 把M_2和签名$H(M_1)$发送给接收者，接收者受到M_2计算其哈希值$H(M_2)=H(M_1)$，完成伪造

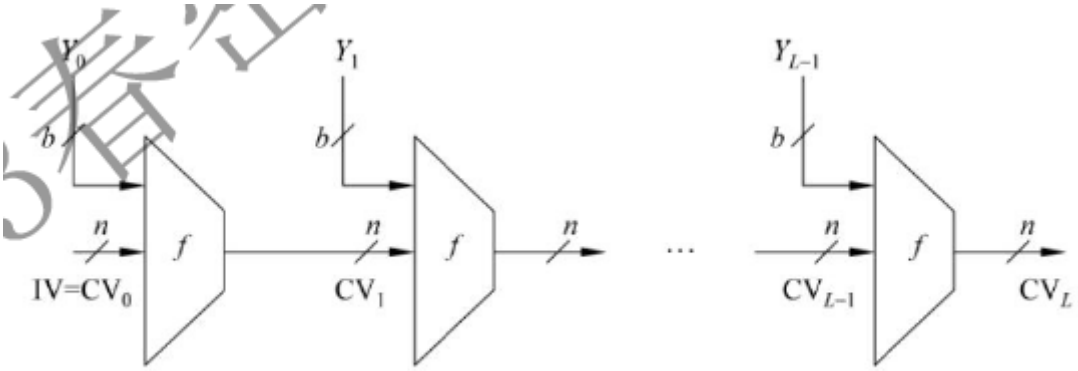
(5)：

1. 基于加密算法的哈希函数构造方法，例如SHA-2系列。这些哈希函数通过将输入块分成512位的消息块，然后使用一个压缩函数对每个消息块进行压缩，最终得到一个256位或更长的哈希值。
2. 基于HMAC的哈希函数构造方法，例如HMAC-SHA-3。它通过在输入消息和密钥的基础上构造一个伪随机函数，并使用该函数对消息进行哈希。
3. 基于散列链的哈希函数构造方法，例如Merkle–Damgard构造。这种构造方法将输入分为若干个块，并将它们链接在一起形成一个散列链。每个块都通过一个压缩函数来更新散列值，并将结果传递给下一个块。

2.

(1): Merkle　　Keccak　　SHA-3　　海绵

(2):



(3): 因为输入的元素集合大小为2^{512}，而输出集合大小为2^{160}，所以一定存在两个相同的输入，输出值相同，即为碰撞。

(4): 任意长度　512bit　128bit　任意长度（但要小于2^{64}bit）　512bit　160bit

(5): MD5和SHA-1的填充都是一样的：

- 在消息x右边增加若干比特，使其长度与448模512同余。也就是说，填充后的消息长度比512的某个倍数少64位。
- 即使消息本身已经满足上述长度要求，仍然需要进行填充。例如，若消息长为448，则仍需要填充512位使其长度为960位。
- 填充位数在1到512之间。填充比特的第一位是1，其它均为0。

SHA-512：

填充消息使其长度模1024与896同余（即长度=896(mod 1024)）即使消息已经满足上述长度要求，仍然需要进行填充，因此填充位数在 1~1024 之间。填充由一个1和后续的0组成。

3.

(1)： 消息认证　密钥控制　　$MAC = C(K, M)$　　输入消息　　共享的密钥

(2)： 用Hash函数构造　　用分组密码构造

(3)：

$I.$

第一轮：已知M_1、$MAC_1 = C_k(M_1)$。对所有2^k个可能的密钥计算$MAC_i = C_{K_i}(M_1)$

第二轮：已知M_2、$MAC_2 = C_K(M_2)$。对上一轮得到的2^{k-n}个可能的密钥计算$MAC_i = C_{K_i}(M_2)$，得到2^{k-2n}个可能的密钥

$II. O(2^k)$

$III.$

MD5算法被认为是不安全的主要原因是它容易受到碰撞攻击，即找到两个不同的消息，使它们具有相同的MD5哈希值。这意味着攻击者可以通过篡改消息来欺骗系统，从而破坏系统的完整性。

HMAC-MD5在MD5哈希函数的基础上引入了一个密钥，使得攻击者很难对哈希值进行任何有效的篡改。这个密钥可以确保只有知道密钥的人才能够正确地计算出哈希值，从而保证了消息的完整性和身份验证。

此外，由于HMAC-MD5算法具有对称密钥的特性，它的计算速度相对较快。