信息安全引论作业 02

梁昱桐 2100013116

Peking University

1 习题 1 (7.4)

在 DES 的 ECB 模式中，若在密文的传输过程中，某一块发生了错误，则只有相应的明文分组会有影响，然而，在 CBC 模式中，这种错误具有扩散性，比如传输时 C_1 发生错误将会影响明文分组 P_1 和 P_2。

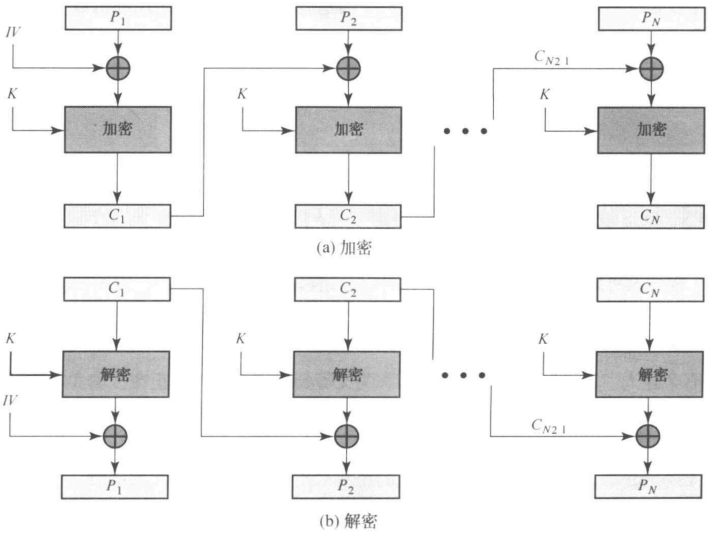


图 7.4 密文分组链接（CBC）模式

1. P_2 以后的分组是否会受到影响?

   会, 所有分组都会

2. 假设 P1 本来就有一位发生了错误，则这个错误要扩散多少个密文分组，对接受者解密后的结果有何影响?

   这个错误会扩展到所有密文分组, 但是不会影响解密结果, 这相当于加密之前的明文从 P_1 变成了 P_1' , 解密的结果依然是正确的.

2 习题 2 (7.6)

CBC-pad 是 RC5 中的一种分组加密工作模式，它可以用于任何分组密码，并处理任意长度的明文，得到的密文最多比明文长一个分组的长度。填充的作用是保证输入明文是分组长度的整数倍。假设原始明文是整数字节，尾部填充 1 到 bb 个字节，这里 bb 为分组的字节长度。填充的字节相同，其值为填充的字节数。例如，若有 8 字节填充，则每个字节为 00001000。那么为何不允许 0 字节的填充？若原始明文是分组大小的整数倍，为什么不省去填充?

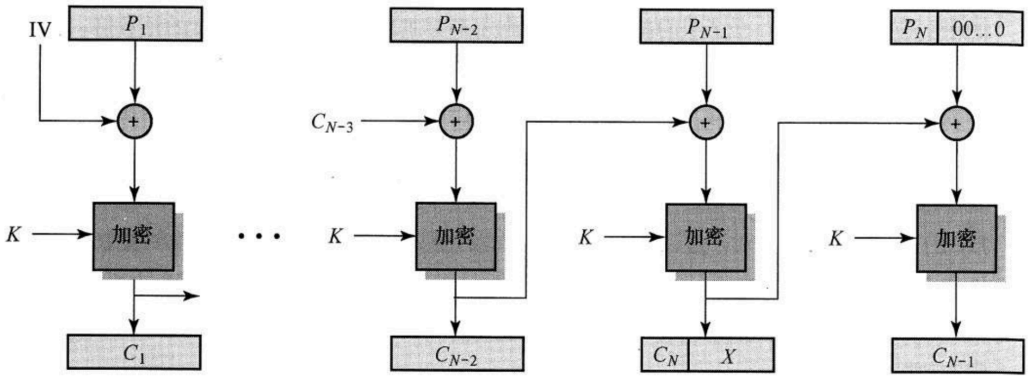
因为在解密时需要明确知道填充的数量，以正确地恢复原始明文。

1. 得知有多少填充: 如果允许0字节填充, 那么在解密过程中无法区分最后一个分组是否包含填充, 包含多少填充, 这也是如果明文是分组大小的整数倍, 仍然需要进行填充的原因.
2. 简化处理: 始终填充至少一个完整分组的字节, 确保解密过程中的处理逻辑简单且一致.
3. 安全性: 如果不添加填充, 可能会导致某些攻击者推断出明文的长度信息.

3 习题 3 (7.11)

填充并不总是合适的，例如，我们希望使用相同的内存缓冲区（明文最初存储在这里）来存储加密的数据，这时密文必须与明文的长度相同。密文挪用模式（CTS）是满足这种要求的一种工作模式。图7.18（a）为该模式的实现过程

图7.18(a)



(a) 密文窃取模式

1. 解释 CTS 是如何工作的;

   假设 n 为分组长度, m 为最后一个不完整分组的长度, 那么倒数第二组密文的长度为 n, 最后一组密文的长度为 m.

   倒数第二次加密得到的是最后一组密文, 最后一组密文 C_n 取倒数第二次加密的密文分组的前 m 个比特位.

   最后一次加密得到的是倒数第二组密文, 对剩余长度 m 的明文进行异或的时候, 对其进行 0 填充到分组长度 n 然后和倒数第二次加密的密文分组进行异或再加密, 得到倒数第二组密文.

2. 描述如何解密 C_{n-1} 和 C_n。

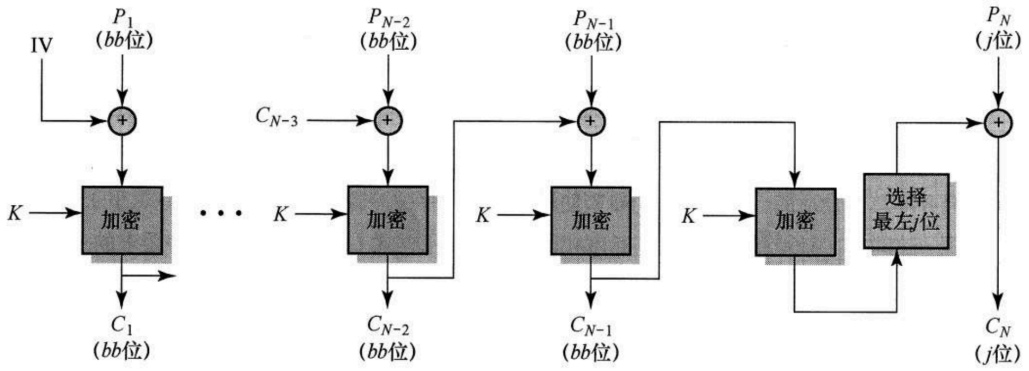
   解密到倒数第二组密文 C_{n-1} 的时候, 解密后, 取前 m 个比特位, 和 C_n 进行异或得到明文 P_n; 后 $n-m$ 个比特位为 X.

   解密最后一组密文 C_n 的时候, 把它和 X 连接起来, 解密后, 和 C_{n-2} 进行异或得到明文 P_{n-1}.

4 习题 4 (7.12)

图7.18(b)中给出了一种 CTS 的替换方案，使得当明文不是分组长度整数倍时产生的密文长度与明文长度相等

图7.18(b)



(b) 替换方法

1. 解释该算法

   在处理最后一个不完整分组的时候, 将它和倒数第二次密文的前 m 个比特位进行异或, 直接得到最后一组密文.

2. 解释为何 CTS 算法比图 7.18(b) 中的方法更可取?

   因为这个算法在处理最后一个不完整分组的时候没有进行加密, 所以最后一组明文有通过测试攻破的可能性.

5 习题 5

分组密码作用于 n 位明文分组，产生 n 位密文分组，当 n=64 或者更大时，为什么使用 n 位分组的任意可逆代替密码不可行?

因为这个可逆变换需要太大的空间存储了.

总共需要存储 2^n 个一一映射, 每个映射需要 n 位存储, 总共需要 $n \times 2^n$ 位存储, 当 $n = 64$ 时, 需要 64×2^{64} 位存储, 这个空间大约为 $10^{21.07}$ 比特.