CBC字节翻转攻击原理

CBC模式：Cipher Black Chaining mode (密码分组链接模式)
CBC模式进行加解密是都需要一个随机初始向量iv，在第一轮进行加解密是都需要与iv进行xor的。

任何字符与本身xor都是为0，任何字符与0xor都为本身，如A xor A=0，A xor 0=A

加密过程
1、将明文分为若干组（16个字节为一组），最后一组不足则用特殊字符填充
2、生成一个初始向量iv和key密钥
3、用iv与第一组明文异或（iv只影响第一组生成的密文）生成密文
4、然后再用前n组密文与后n+1组明文异或生成第n+1组密文，以次重复
5、最后将生成的密文拼接起来，就成了最终密文
加密公式：
Ciphertext-0 = Encrypt(Plaintext XOR IV)—只用于第一个组块
Ciphertext-N= Encrypt(Plaintext XOR Ciphertext-N-1)—用于第二及剩下的组块

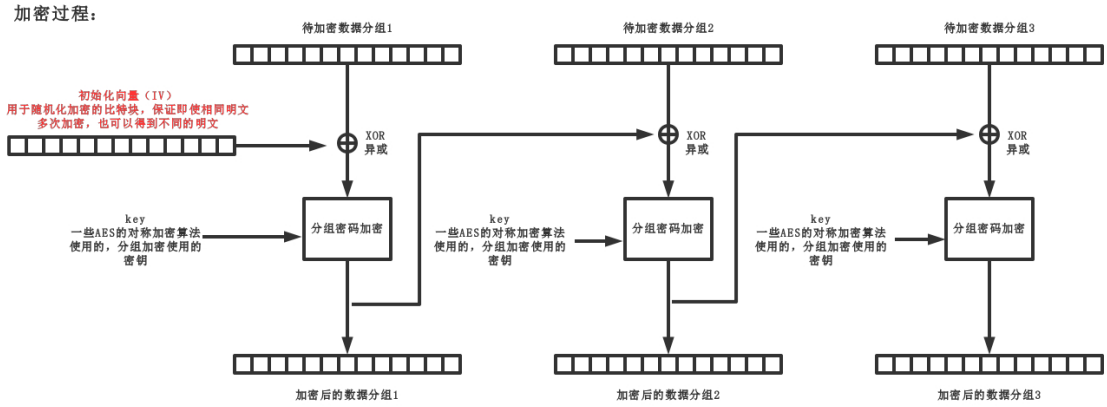
解密过程：
1、将密文分组
2、用iv与第一组密文xor，解密得到第一组明文
3、用第n组密文与第n+1组密文xor，解密得到第n+1组明文，以此类推
4、将各组的明文拼接在一起就是最终要得到的明文了

注意一下：解密的时候前一组密文只影响后一组明文的结果，而不会影响其他组明文的结果，由图也可看得出，这个也是进行攻击的重要之处。

有一条经验法则是（注：结合上面的说明图可以得到），你在密文中改变的字节，只会影响到在下一明文当中，具有相同偏移量的字节。所以我们目标的偏移量是2：

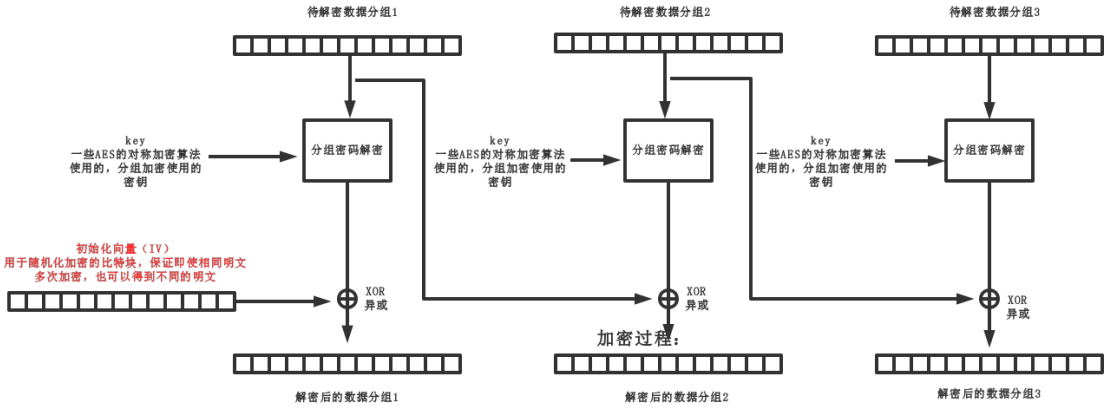
CBC字节翻转攻击原理图

加密过程：



1. 将明文分为若干组（16个字节为一组），最后一组不足则用特殊字符填充
2. 生成一个初始向量iv和key密钥
3. 用iv与第一组明文异或（iv只影响第一组生成的密文）生成密文
4. 然后再用前n组密文与后n+1组明文异或生成第n+1组密文，以次重复
5. 最后将生成的密文拼接起来，就成了最终密文

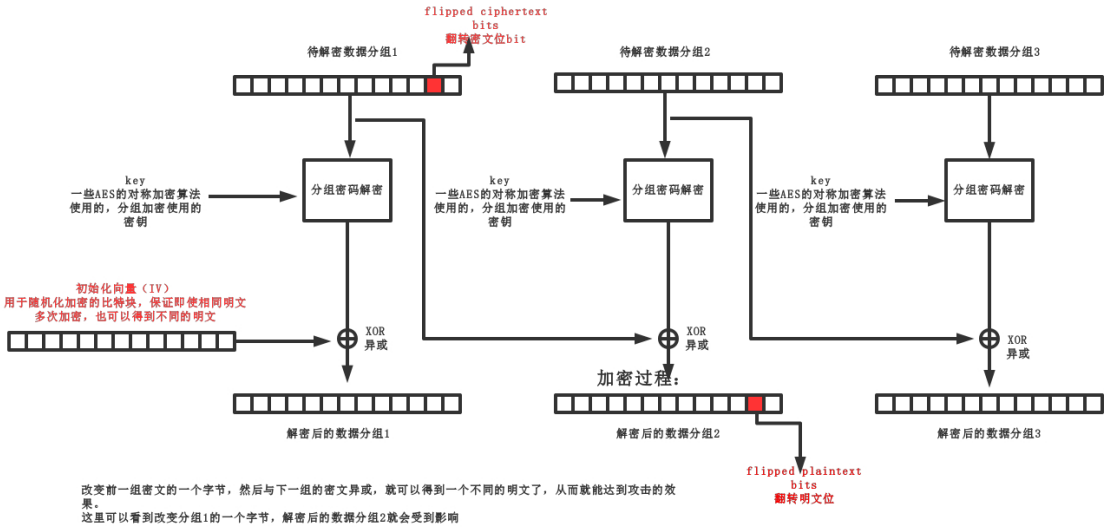
解密过程：



1. 将密文分组
2. 用iv与第一组密文xor，解密得到第一组明文
3. 用第n组密文与第n+1组密文xor，解密得到第n+1组明文，以此类推
4. 将各组的明文拼接在一起就是最终要得到的明文

注意：解密的时候前一组密文只影响后一组明文的结果，而不会影响其他组明文的结果，由图也可看得出，这个也是进行攻击的重要之处。

CBC字节翻转攻击：



改变前一组密文的一个字节，然后与下一组的密文异或，就可以得到一个不同的明文了，从而就能达到攻击的效果。
这里可以看到改变分组1的一个字节，解密后的数据分组2就会受到影响

小维同学
www.axiaowei.cn

http://processon.com/chart_image/5c37fd8de4b0db2e59306630.png

XOR异或运算
异或，英文为exclusive OR，缩写成xor
异或的数学符号为“⊕”
异或略称为XOR、EOR、EX-OR
程序中有三种演算子：XOR、xor、⊕。
两个输入相同时为0，不同则为1