分组对称加密模式:ECB/CBC/CFB/OFB

How can AES modes help the decryption overheads?

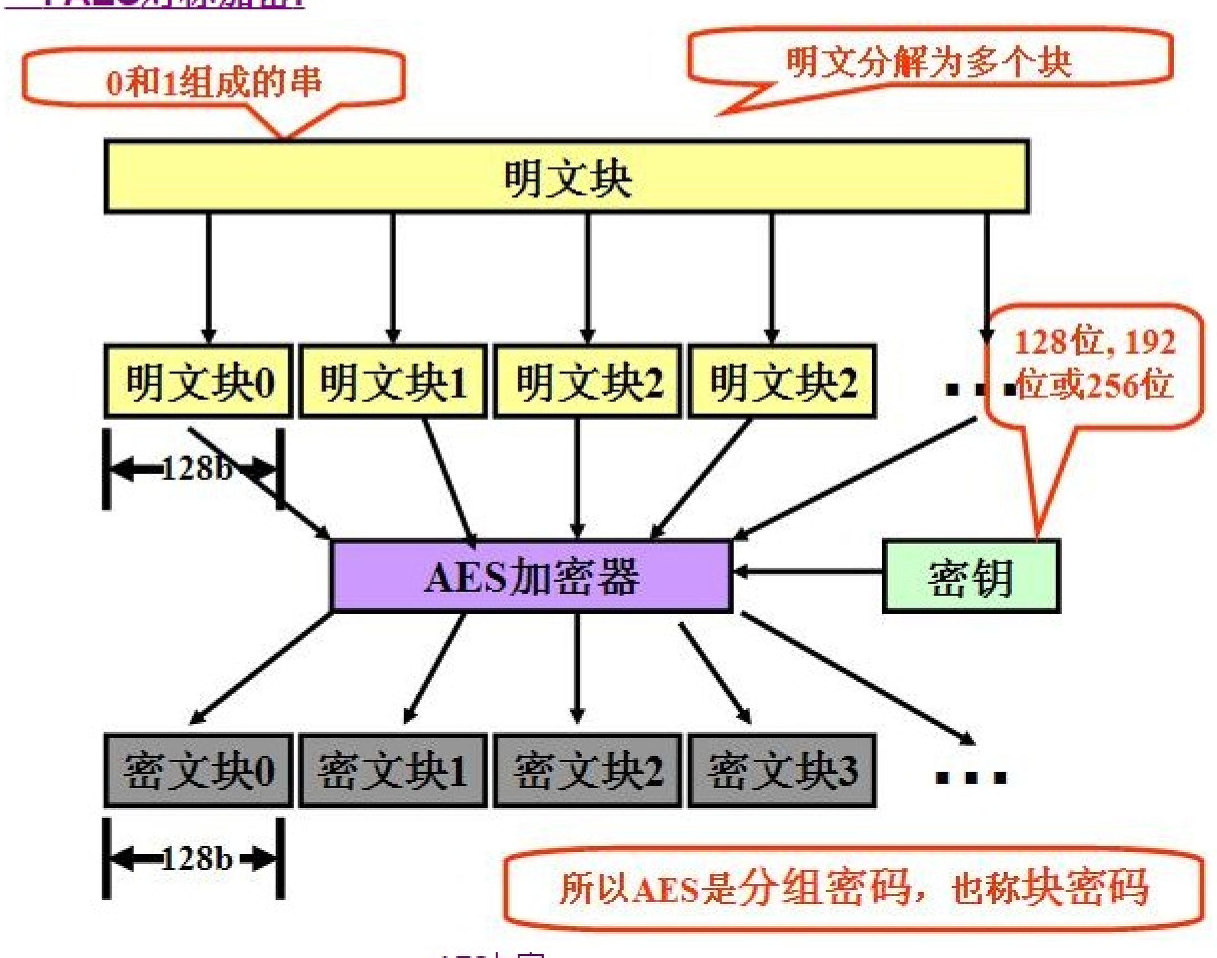
Some of the common AES modes do not require an ECB decryptor in order to perform decryption! For example, due to the way in which the feedback is structured, CFB, OFB and CTR modes use an AES ECB Encryptor in both their encrypt and decrypt functions. This is especially useful since the AES ECB decryptor, as detailed previously, has several fundamental disadvantages.

Using an encryptor at both ends of the system, allows a perfectly balanced system to be attained, running at the highest possible rate, without suffering the bottleneck of using a decryptor with its potential speed, size and key update issues. This is something the original designers of Rijndael were fully aware of, and cite as a reason for accepting the complications of their ECB decryption scheme.

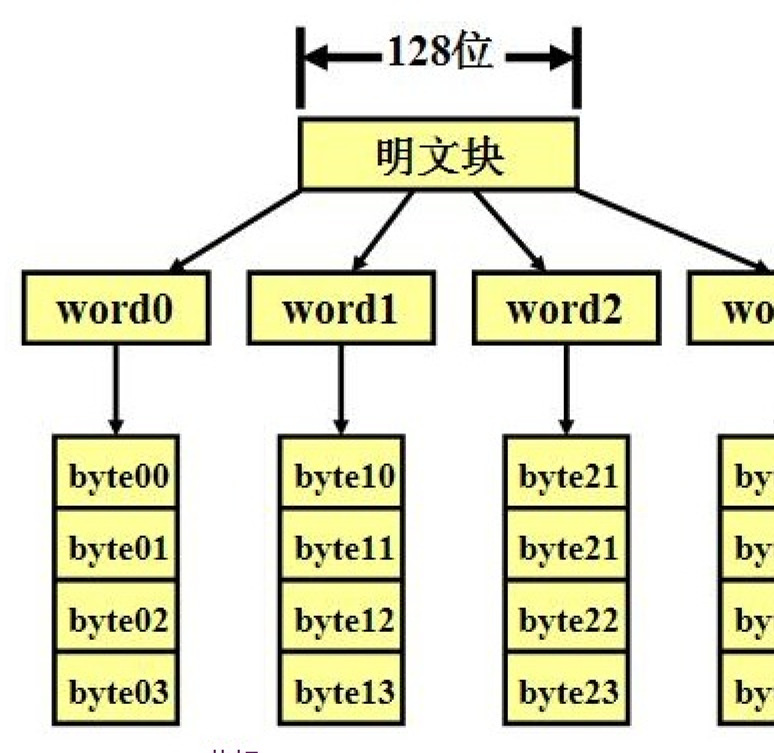
一般的加密通常都是块加密，如果要加密超过块大小的数据，就需要涉及填充和链加密模式，文中提到的ECB和CBC等就是指链加密模式。在C#组件中实现的很多算法和Java都不太兼容，至少我发现RSA和AES/ECB是如此。研究了AES/ECB时发现了这篇文档，图还画的不错，先记下。注意，还缺一种CTR的模式。

对称加密和分组加密中的四种模式(ECB、CBC、CFB、OFB)

一. AES对称加密:

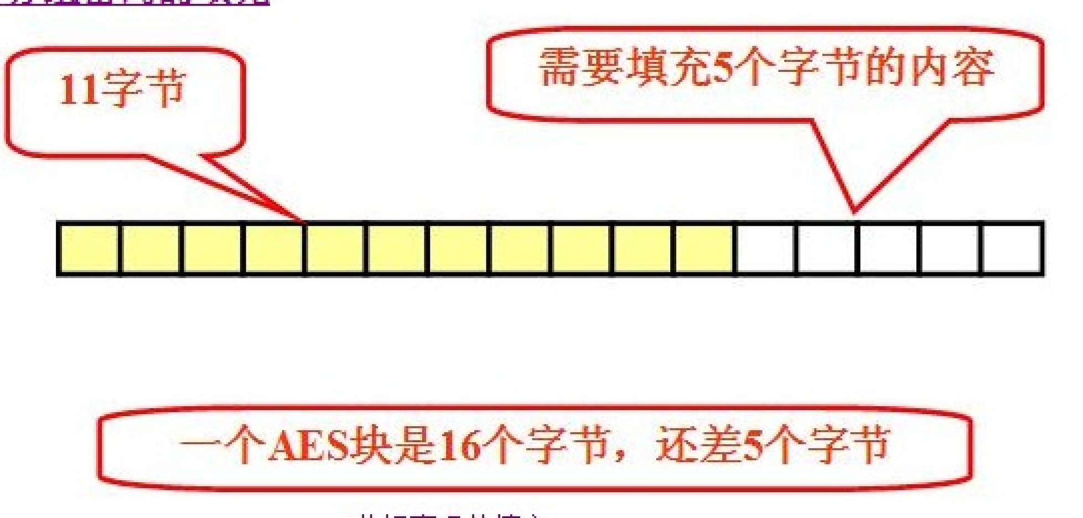


AES加密



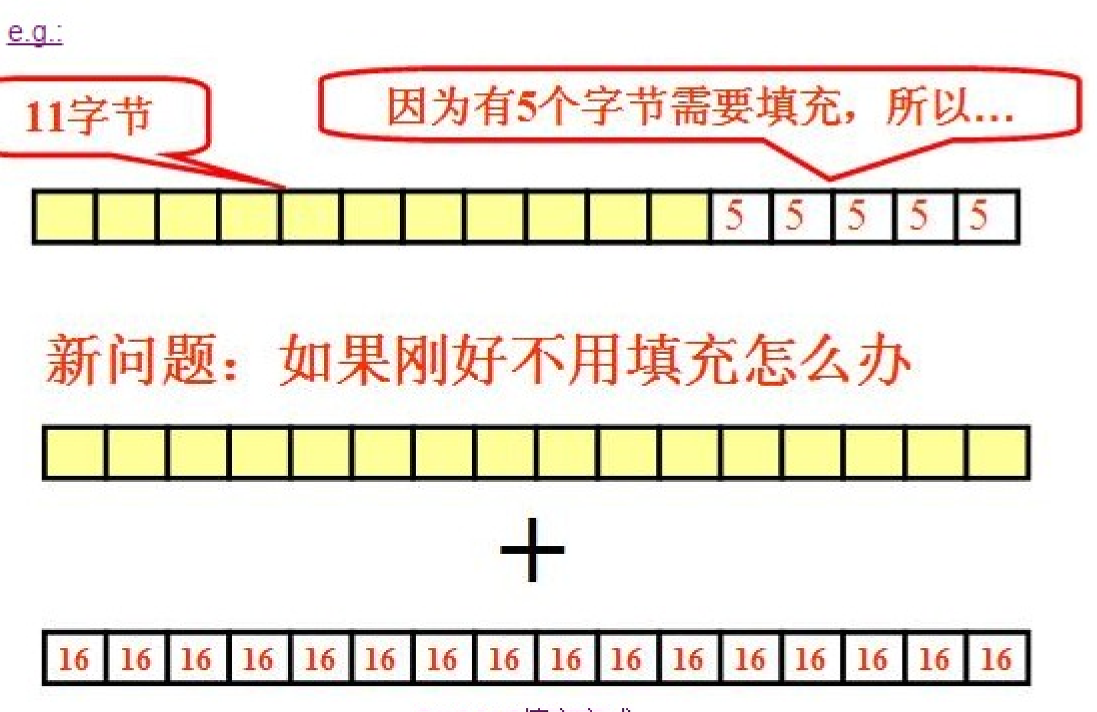
分组

二. 分组密码的填充



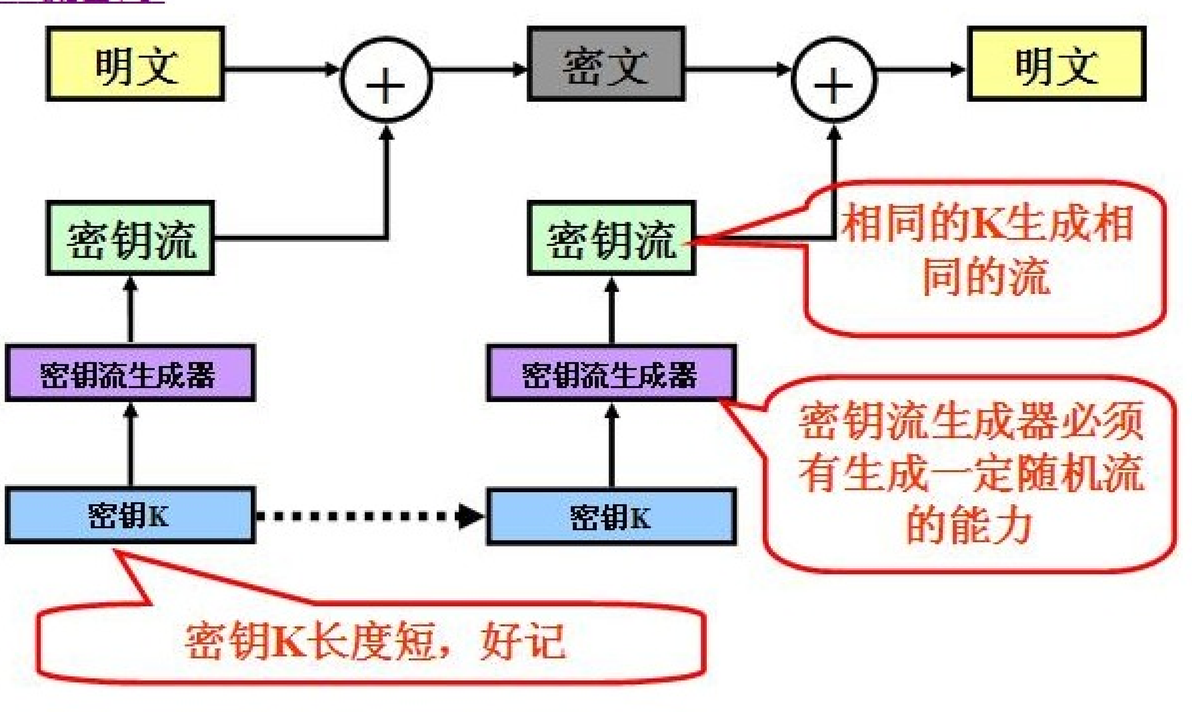
分组密码的填充

e.g.:



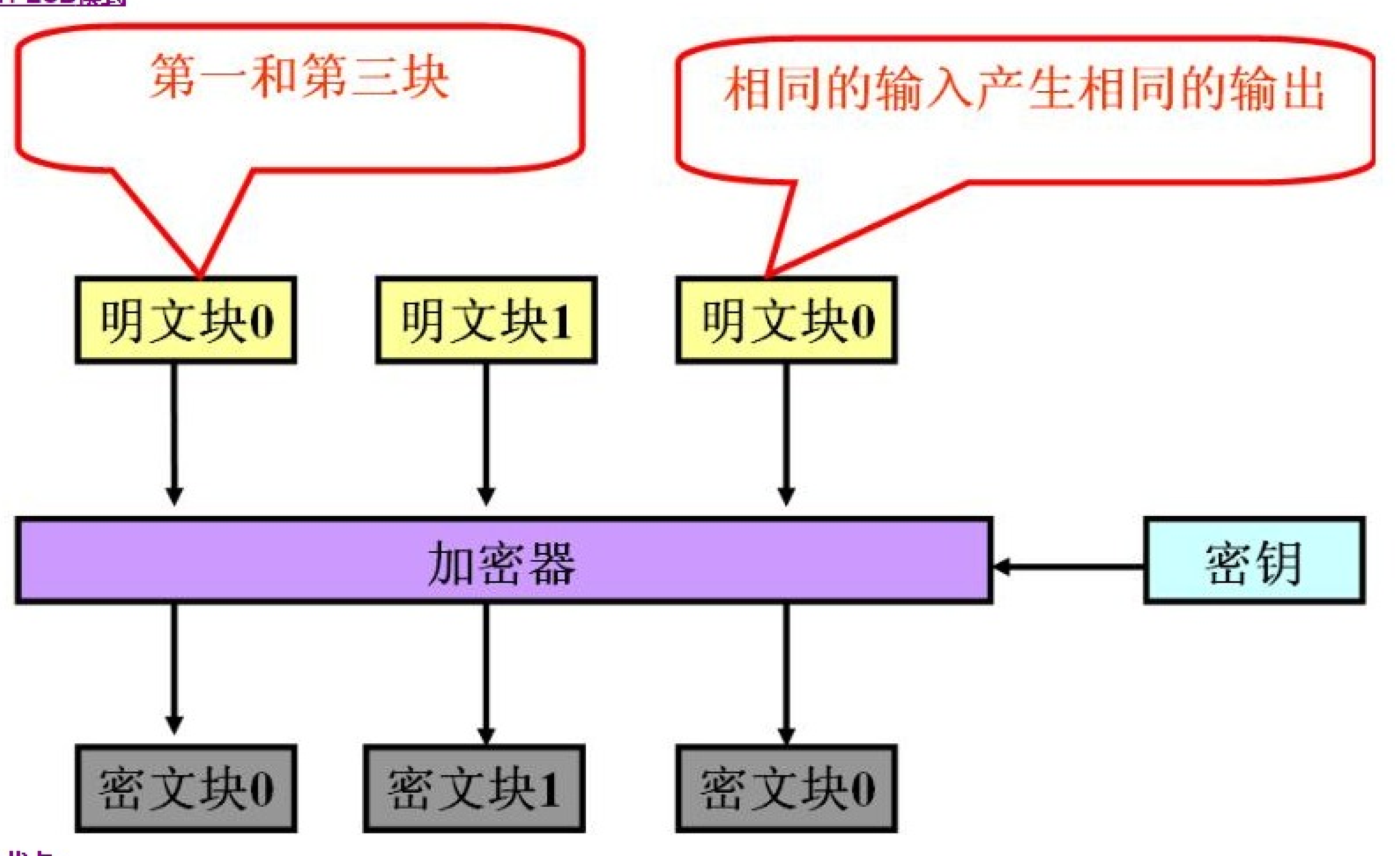
PKCS#5填充方式

三. 流密码:



四. 分组密码加密中的四种模式:

3.1 ECB模式



优点:

1.简单；

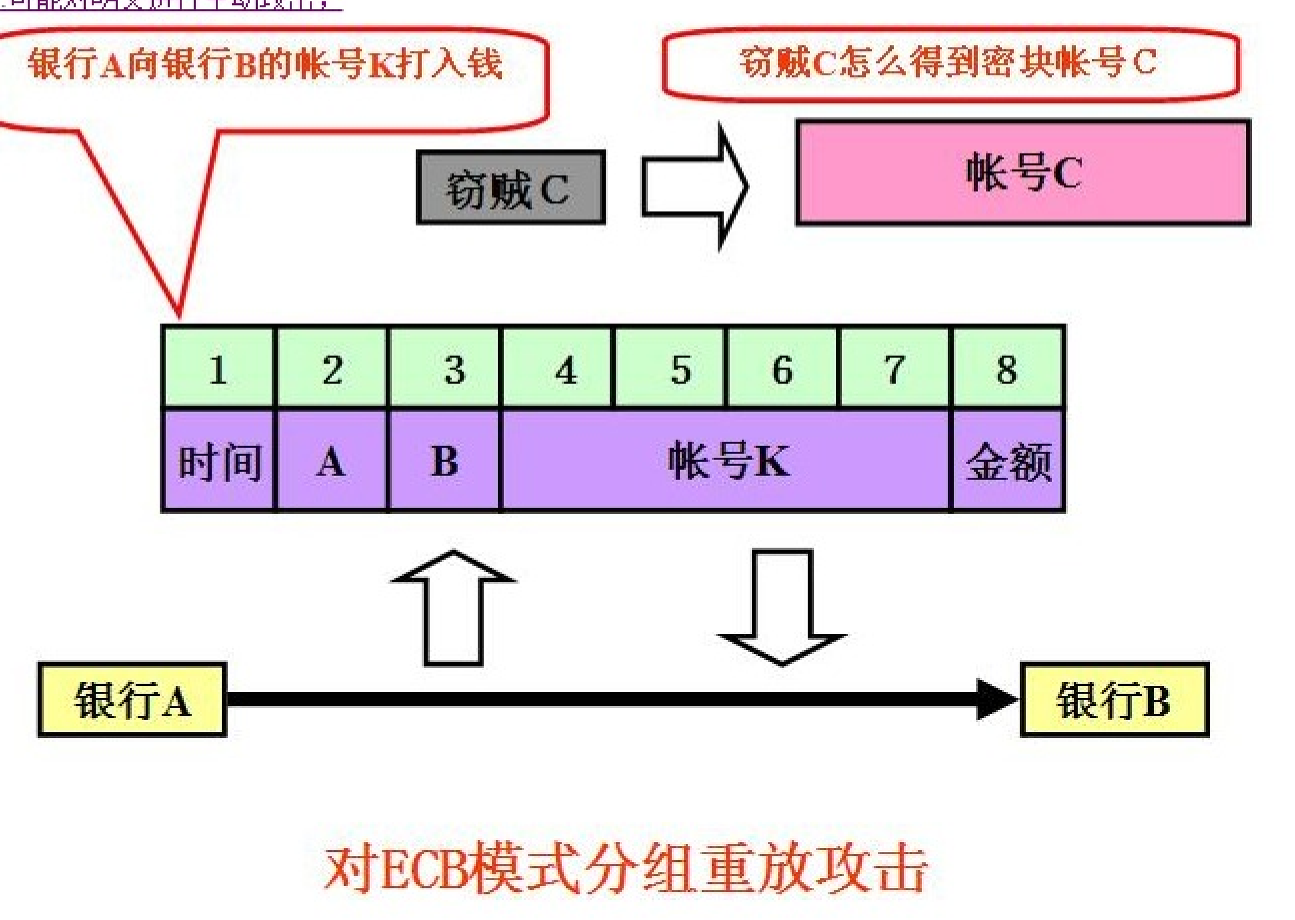
2.有利于并行计算；

3.误差不会被传送；

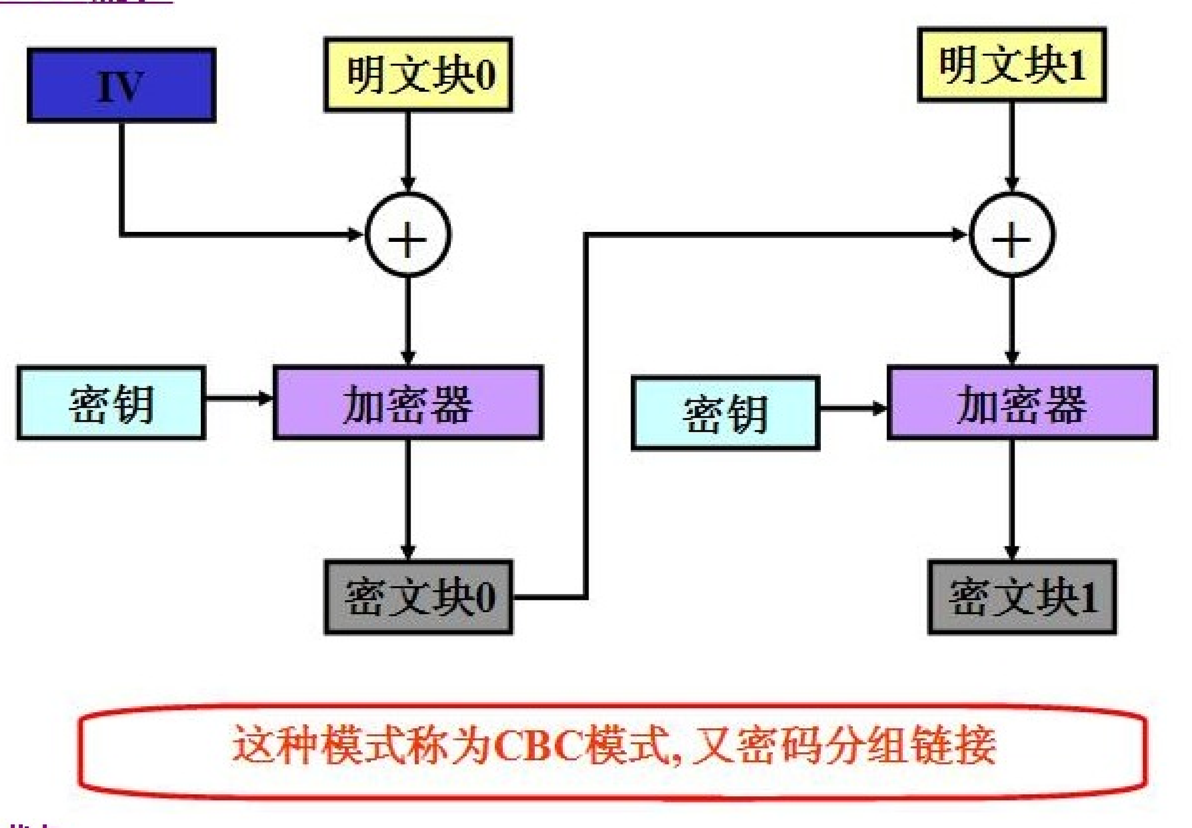
缺点:

1.不能隐藏明文的模式；

2.可能对明文进行主动攻击；



3.2 CBC模式：



优点：

1.不容易主动攻击,安全性好于ECB,适合传输长度长的报文,是SSL、IPSec的标准。

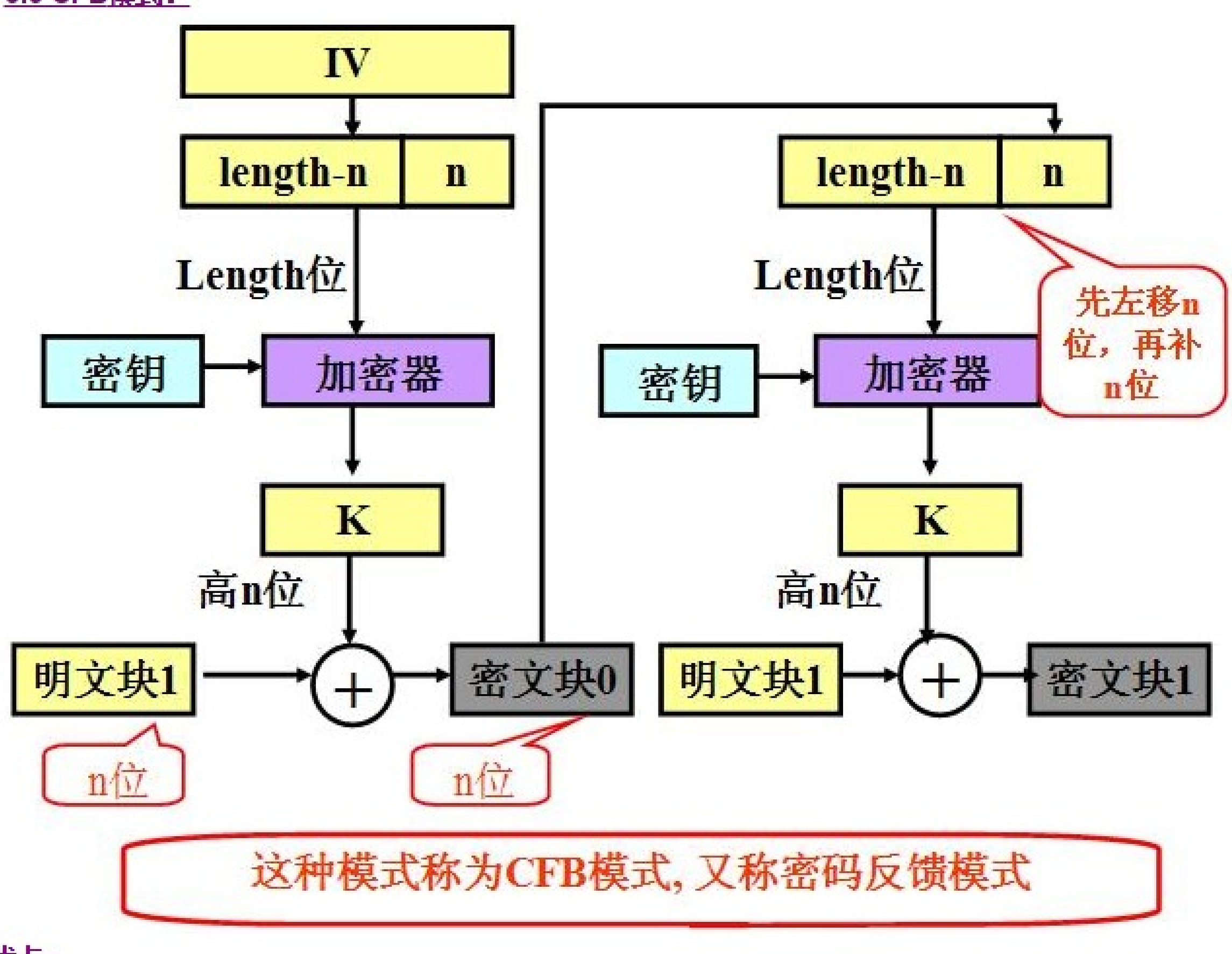
缺点：

1.不利于并行计算；

2.误差传递；

3.需要初始化向量IV

3.3 CFB模式：



优点：

1.隐藏了明文模式;

2.分组密码转化为流模式;

3.可以及时加密传送小于分组的数据;

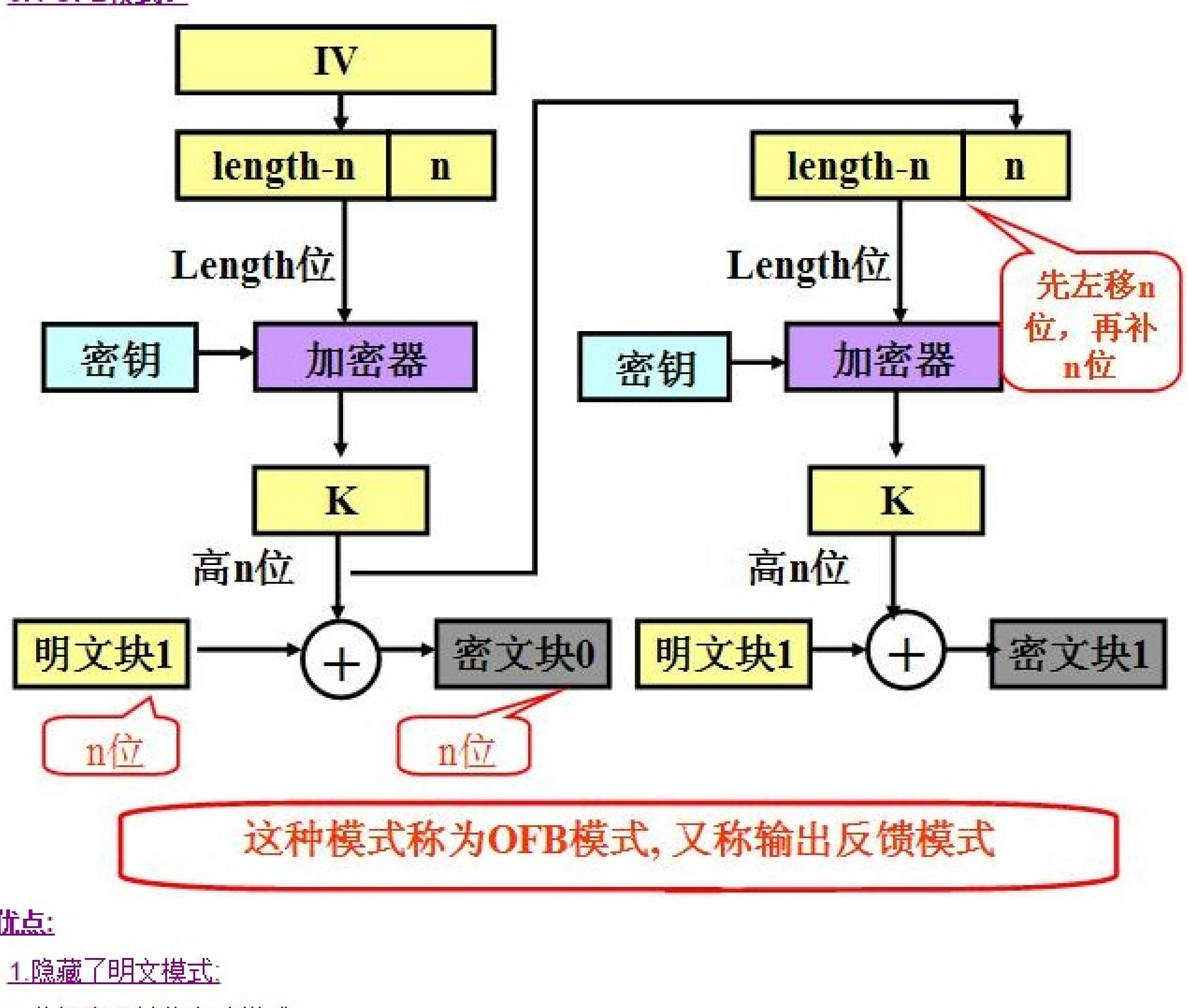
缺点:

1.不利于并行计算;

2.误差传送：一个明文单元损坏影响多个单元;

3.唯一的IV;

3.4 OFB模式：



优点:

1.隐藏了明文模式;

2.分组密码转化为流模式;

3.可以及时加密传送小于分组的数据;

缺点:

1.不利于并行计算;

2.对明文的主动攻击是可能的;

3.误差传送：一个明文单元损坏影响多个单元;