的密文块。为了解释这个想法，令 $m(i)$ 表示第 i 个明文块，$c(i)$ 表示第 i 个密文块，并且 $a \oplus b$ 表示两个比特串 a 和 b 的异或（XOR）。（前面讲过 $0 \oplus 0 = 1 \oplus 1 = 0$ 和 $0 \oplus 1 = 1 \oplus 0 = 1$，并且两个比特串的异或是逐位进行的。因此有，例如 10101010 ⊕ 11110000 = 01011010。）另外，将具有密钥 S 的块密码加密算法表示为 K_S。其基本思想如下：发送方为第 i 块生成一个随机的 k 比特数 $r(i)$，并且计算 $c(i) = K_S(m(i) \oplus r(i))$。注意到每块选择一个新的 k 比特随机数。则发送方发送 $c(1)$、$r(1)$、$c(2)$、$r(2)$、$c(3)$ 和 $r(3)$ 等等。因为接收方接收到 $c(i)$ 和 $r(i)$，它能够通过计算 $m(i) = K_S(c(i) \oplus r(i))$ 而恢复每个明文块。重要的是注意到下列事实：尽管 $r(i)$ 是以明文发送的，并且因此能被 Trudy 嗅探到，但她无法获得明文 $m(i)$，因为她不知道密钥 K_S。同时注意到如果两个明文块 $m(i)$ 和 $m(j)$ 是相同的，对应的密文块 $c(i)$ 和 $c(j)$ 将是不同的（只要随机数 $r(i)$ 和 $r(i)$ 不同，这种情况出现的概率将很高）。

举例来说，考虑在表 8-1 中的 3 比特块密码。假设明文是 010010010。如果 Alice 直接对此加密，没有包括随机性，得到的密文变为 101101101。如果 Trudy 嗅探到该密文，因为这三个密文块的每个都是相同的，她能够正确地推断出这三个明文块的每个都是相同的。现在假设 Alice 产生了随机块 $r(1) = 001$、$r(2) = 111$ 和 $r(3) = 100$，并且使用了上述技术来生成密文 $c(1) = 100$、$c(2) = 010$ 和 $c(3) = 000$。注意到即使明文块相同，三个密文块也是不同的。Alice 则发送 $c(1)$、$r(1)$、$c(2)$ 和 $r(2)$。读者可证实 Bob 能够使用共享的密钥 K_S 获得初始的明文。

机敏的读者将注意到，引入随机性解决了一个问题而产生了另一个问题：Alice 必须传输是以前两倍的比特。实际上，对每个加密比特，她现在必须再发送一个随机比特，使需要的带宽加倍。为了有效利用该技术，块密码通常使用了一种称为**密码块链接**（Cipher Block Chaining，CBC）的技术。其基本思想是仅随第一个报文发送一个随机值，然后让发送方和接收方使用计算的编码块代替后继的随机数。具体而言，CBC 运行过程如下：

1）在加密报文（或数据流）之前，发送方生成一个随机的 k 比特串，称为**初始向量**（Initialization Vector，IV）。将该初始向量表示为 $c(0)$。发送方以明文方式将 IV 发送给接收方。

2）对第一个块，发送方计算 $m(1) \oplus c(0)$，即计算第一块明文与 IV 的异或。然后通过块密码算法运行得到的结果以得到对应的密文块，即 $c(1) = K_S(m(1) \oplus c(0))$。发送方向接收方发送加密块 $c(1)$。

3）对于第 i 个块，发送方根据 $c(i) = K_S(m(i) \oplus c(i-1))$ 生成第 i 个密文块。

我们现在来考察这种方法的某些后果。首先，接收方将仍能够恢复初始报文。毫无疑问，当接收方接收到 $c(i)$，它用 K_S 解密之以获得 $s(i) = m(i) \oplus c(i-1)$；因为接收方已经知道 $c(i-1)$，它则从 $m(i) = s(i) \oplus c(i-1)$ 获得明文块。第二，即使两个明文块是相同的，相应的密文块将（几乎）总是不同的。第三，虽然发送方以明文发送 IV，入侵者将仍不能解密密文块，因为该入侵者不知道秘密密钥 S。最后，发送方仅发送一个最前面的块（即 IV），因此对（由数百块组成的）长报文而言增加的带宽用量微不足道。

举例来说，对表 8-1 中的 3 比特块密码，明文为 010010010 和 $IV = c(0) = 001$，我们现在来确定其密文。发送方首先使用 IV 来计算 $c(1) = K_S(m(1) \oplus c(0)) = 100$。发送方然后计算 $c(2) = K_S(m(2) \oplus c(1) = K_S(010 \oplus 100) = 000$，并且 $c(3) = K_S(m(3) \oplus c(2) =$