

图 E-1 是一个简单的 Base-64 编码实例。在这里，三个字符组成的输入值“Ow!”是 Base-64 编码的，得到的是 4 个字符的 Base-64 编码值“T3ch”。它是按以下方式工作的。

- (1) 字符串“Ow!”被拆分成 3 个 8 位的字节 (0x4F、0x77、0x21)。
- (2) 这 3 个字节构成了一个 24 位的二进制值 0100111110111011100100001。
- (3) 这些位被划分为一些 6 位的序列 010011、110111、01110、100001。
- (4) 每个 6 位值都表示从 0 ~ 63 之间的一个数字，对应 Base-64 字母表中 64 个字符之一。得到的 Base-64 编码字符串是个 4 个字符的字符串“T3ch”，然后就可以通过线路将这个字符串作为“安全的”8 位字符传送出去，因为只用了一些移植性最好的字符（字母、数字等）。

8位字符	0	w	!	
8位值 (十六进制)	\$4F	\$77	\$21	
8位值 (二进制)	010011110111011100100001			
6位值 (十进制)	19	55	28	33
Base-64字符	T	3	c	h

图 E-1 Base-64 编码实例

571

E.3 Base-64填充

Base-64 编码收到一个 8 位字节序列，将这个二进制序列流划分成 6 位的块。二进制序列有时不能正好平均地分成 6 位的块，在这种情况下，就在序列末尾填充零位，使二进制序列的长度成为 24 的倍数（6 和 8 的最小公倍数）。

对已填充的二进制串进行编码时，任何完全填充（不包含原始数据中的位）的 6 位组都由特殊的第 65 个符号“=”表示。如果 6 位组是部分填充的，就将填充位设置为 0。

表 E-2 显示了一些填充实例。初始输入字符串“a:a”为 3 字节（24 位）。24 是 6 和 8 的倍数，因此无需填充，得到的 Base-64 编码字符串为“YTph”。