**文件密钥生成算法**

# 说明

当前的加密算法中，代码密钥是固化在程序中的，而文件密钥是每次加密文件的时候随机生成的。

这导致每次加密后的文件内容都不一致，如果每次打包的时候进行文件加密操作，将无法实现差分升级的功能。

为了优化这个问题，设计一种算法，对于每个文件，每次产生的文件密钥是相同的，这样无论何时执行文件加密，只要使用相同的代码密钥，得到的加密文件是一样的。

# 原有文件密钥规范

原有的文件密钥是一个二进制字节数组。

字节数组长度在16-32之间随机生成。

字节数组内容是随机生成。

# 新文件密钥规范

新文件密钥也是一个二进制字节数组，与原有规范兼容，解密算法不用修改。

密钥长度=文件长度%17+16

例如：

文件长度=1000

密钥长度=1000%17+16=14+16=30

密钥长度为30

密钥内容计算方式如下：

1、先初始化为0

2、依次与文件内容进行异或xor计算

3、当字节数组到末尾时，重新移动到数组头部，进行计算

4、直到文件内容结束。

例如：

文件长度为38个字节，内容用数字表示为｛1，2，3，4，5，6，7，8，……，38｝

则密钥长度=38%17+16=20

1、初始化为0，密钥数组=｛0，0，0，0，……，0｝，长度=20

2、第一遍异或，密钥数组=｛1，2，3，4，……，20｝，长度=20

3、第二遍循环，

密钥数组=｛1 xor 21，2 xor 22，3 xor 23，……，18 xor 38，19，20｝，长度=20

密钥数组=｛20，20，20，……，52，19，20｝，长度=20

4、文件内容结束

所以密钥数组为｛20，20，20，……，52，19，20｝，长度=20