一种文件加密与完整性验证工具

**用法**

xcp [选项] 文件

xcp [选项] 源文件 目标文件

xcp [选项] 源文件 目录

xcp [选项] 源目录 目标目录

路径是否以路径分割符(Linux下'/', Windows下'\')结束，没有任何影响。

将源文件或多个源文件复制到目标目录, 如果存在同名文件，则在权限允许的情况下覆盖；

如果目标目录存在，则将源目录拷贝到目标目录下，保留原目录名；否则创建目标目录，并将源目录下的所有文件以及子目录拷贝到目标目录下；

xcp [-u] 源文件 目标文件

xcp [-u] -e|--encrypt –k=pkey|--key=pkey|--key pkey 源文件 加密后文件

xcp [-u] -d|--decrypt –k=pkey|--key=pkey|--key pkey 加密文件 解密后文件

xcp -c|--check –k=pkey|--key=pkey|--key pkey 文件或目录

xcp --md5sum 文件或目录

在进行加密解密校验操作时，如果未指定密匙则采用工具默认密匙进行操作；

进行文件加密操作时，如果目标文件后缀名不为.cxc，则为目标文件追加后缀名.cxc;

进行文件解密操作时，只对后缀名为.cxc的文件进行校验和解密操作，并去除.cxc后缀名；

进行文件校验操作时，只校验后缀名为.cxcx的文件；

**加密文件结构**

文件头:

16 Bytes : pkey追加源文件加密后内容的md5校验码(16位长);

1 Bytes : 源文件字节数对16的模数;

文件data域:

16 x N Bytes : 源文件根据AES算法加密后的内容

**命令含参数的解析**

**配置参数：**采用getopt\_long函数实现；

**路径信息参数：**进行拷贝相关操作时(拷贝，加密，解密)，要求至少两个路径信息参数；最后一个路径参数作为目标路径处理；

**文件通配符的展开**

Linux环境下，该功能有shell处理；

Windows环境下，需要程序处理，调用API FindFirstFile，FindNextFile实现；

**加密解密**

采用AES 16加密算法进行加密解密处理；

以16字节为单位，对源文件内容进行加密，不足16字节的内容加密后仍未16字节，将不足16字节的长度存入加密后文件头部；

以16字节为单位，对加密文件内容进行解密，文件最后16字节需要特殊处理，从加密文件头部读出该部分加密前的字节数；

**加密文件校验**

加密时，以密匙追加加密后文件主体内容的MD5码，共16字节，写到加密文件头部前16字节；

检验时，进行类似操作，如果计算出的16位校验和与文件头前16字节相同，则校验成功，否则校验失败；

**目录遍历**

Linux：opendir(),readdir();

Windows：FindFirstFile()，FindNextFile();

**文件拷贝操作**

以文件A拷贝到B为例；

一般拷贝操作：

如果B为文件，直接拷贝；

如果B为目录，B必须存在，目标文件为B/A；

加密操作：

如果B为文件，并且后缀名为.cxc，加密拷贝；

如果B为文件，但后缀名不为.cxc，加密拷贝到B.cxc；

如果B为目录，B必须存在，加密拷贝到B/A.cxc；

解密操作：

如果A后缀名不为.cxc，放弃；

如果B为文件，解密拷贝；

如果B为目录，A’为A去除.cxc后缀名的文件名，解密拷贝到B/A’；

**程序源码**

AES库：aes.h，aes.c；

MD5库：md5.h，md5.c；

操作系统相关：os.h，判断编译平台，以及平台相关定义；

func.c，func.h：功能函数，创建文件夹，获取文件名，后缀名，获取文件追后修改时间等函数实现；

xcp.c，xcp.h：xcp主实现文件；

main.c：命令行参数处理，文件通配符展开；