开发Linux系统下的磁盘加密方法详解 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/220/2021_2022__E5_BC_80_E5_8F_91Linu_c103_220866.htm 随着智能手机的计算能力和存储能力的提高，手机中将会存放越来越多的私有数据，这些数据的泄密可能造成严重后果。手机信息安全一直是我们的重点之一，对于一些重要的功能我们要求鉴权后才能使用，但这只能挡住初级的黑客，只能防君子不能防小人，所以我们希望把重要的数据进行加密后再保存。为此，今天花了一点时间去了解Linux 磁盘加密的方法。 方法一：cryptoloop 下载并编译util-linux http://www.paranoiacs.org/~sluskyb/hacks/util-linux/losetup-combined.patch http://ftp.cwi.nl/aeb/util-linux/util-linux-2.12.tar.gz http://hydra.azilian.net/util-linux-2.12-kernel-2.6.patch tar zxvf util-linux-2.12.tar.gz cd util-linux-2.12 patch -p1 patch -p1 (如果有_syscall5之类编译错误，将它换成新的调用方式syscall) make.make install 编译内核(已经支持cryptoloop则跳过此步) make menuconfig Device Drivers >Block Devices>Loopback device support BLK_DEV_CRYPTOLOOP 加载模块 modprobe cryptoloop (以及加密模块) 创建loop设备 dd if=/dev/zero of=~/cryptoloop.image bs=1M count=10 losetup -e aes-256 /dev/loop0 ~/cryptoloop.image (提示输入密码) 创建文件系统并加载 mkfs.ext3 /dev/loop0 mkdir /mnt/crypto mount -t ext3 ~/cryptoloop.image /mnt/crypto/ -oencryption=aes-256 (提示输入密码) 卸载 umount /mnt/crypto losetup -d /dev/loop0 重新加载 losetup -e aes-256 /dev/loop0 ~/cryptoloop.image mount -t ext3

~/cryptoloop.image /mnt/crypto/ -oencryption=aes-256 cryptoloop的实现比较简单，可以看看drivers/block/cryptoloop.c中的代码。loop设备在读写之前会调用lo_do_transfer函数，该函数再调用所安装的transfer插件。cryptoloop就是一种transfer的实现。至于使用哪种transfer及transfer的参数(如密码)，这可以通过LOOP_SET_STATUS64的ioctrl系统调用来完成(mount命令就是这样实现的)。 cryptoloop的缺点是只能针对loop设备，而且对日志型文件系统无效。 