Author: Andreas Steinmetz <ast@domdv.de>

How to use dm-crypt and swsusp together:

===

Some prerequisites:
You know how dm-crypt works. If not, visit the following web page:
http://www.saout.de/misc/dm-crypt/
You have read Documentation/power/swsusp.txt and understand it.
You did read Documentation/initrd.txt and know how an initrd works.
You know how to create or how to modify an initrd.

Now your system is properly set up, your disk is encrypted except for the swap device(s) and the boot partition which may contain a mini system for crypto setup and/or rescue purposes. You may even have an initrd that does your current crypto setup already.

At this point you want to encrypt your swap, too. Still you want to be able to suspend using swsusp. This, however, means that you have to be able to either enter a passphrase or that you read the key(s) from an external device like a pcmcia flash disk or an usb stick prior to resume. So you need an initrd, that sets up dm-crypt and then asks swsusp to resume from the encrypted swap device.

The most important thing is that you set up dm-crypt in such a way that the swap device you suspend to/resume from has always the same major/minor within the initrd as well as within your running system. The easiest way to achieve this is to always set up this swap device first with dmsetup, so that it will always look like the following:

```
brw-------  1 root root 254, 0 Jul 28 13:37 /dev/mapper/swap0
```

Now set up your kernel to use /dev/mapper/swap0 as the default resume partition, so your kernel .config contains:

```
CONFIG_PM_STD_PARTITION="/dev/mapper/swap0"
```

Prepare your boot loader to use the initrd you will create or modify. For lilo the simplest setup looks like the following lines:

```
image=/boot/vmlinuz
initrd=/boot/initrd.gz
label=linux
append="root=/dev/ram0 init=/linuxrc rw"
```

Finally you need to create or modify your initrd. Lets assume you create an initrd that reads the required dm-crypt setup from a pcmcia flash disk card. The card is formatted with an ext2 fs which resides on /dev/hde1 when the card is inserted. The card contains at least the encrypted swap setup in a file named "swapkey". /etc/fstab of your initrd contains something like the following:

```
/dev/hda1   /mnt    ext3      ro                            0 0
none        /proc   proc      defaults,noatime,nodiratime   0 0
none        /sys    sysfs     defaults,noatime,nodiratime   0 0
```

/dev/hda1 contains an unencrypted mini system that sets up all of your crypto devices, again by reading the setup from the pcmcia flash disk. What follows now is a /linuxrc for your initrd that allows you to resume from encrypted swap and that continues boot with your mini system on /dev/hda1 if resume does not happen:

```
#!/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
mount /proc
mount /sys
mapped=0
noresume=`grep -c noresume /proc/cmdline`
if [ "$*" != "" ]
then
  noresume=1
fi
dmesg -n 1
/sbin/cardmgr -q
for i in 1 2 3 4 5 6 7 8 9 0
do
  if [ -f /proc/ide/hde/media ]
  then
    usleep 500000
    mount -t ext2 -o ro /dev/hde1 /mnt
    if [ -f /mnt/swapkey ]
    then
      dmsetup create swap0 /mnt/swapkey > /dev/null 2>&1 && mapped=1
    fi
    umount /mnt
    break
  fi
  usleep 500000
done
killproc /sbin/cardmgr
dmesg -n 6
if [ $mapped = 1 ]
then
  if [ $noresume != 0 ]
  then
    mkswap /dev/mapper/swap0 > /dev/null 2>&1
  fi
  echo 254:0 > /sys/power/resume
  dmsetup remove swap0
fi
umount /sys
mount /mnt
umount /proc
cd /mnt
pivot_root . mnt
mount /proc
```

```
umount -l /mnt
umount /proc
exec chroot . /sbin/init $* < dev/console > dev/console 2>&1
```

Please don't mind the weird loop above, busybox's msh doesn't know the let statement. Now, what is happening in the script?
First we have to decide if we want to try to resume, or not.
We will not resume if booting with "noresume" or any parameters for init like "single" or "emergency" as boot parameters.

Then we need to set up dmcrypt with the setup data from the pcmcia flash disk. If this succeeds we need to reset the swap device if we don't want to resume. The line "echo 254:0 > /sys/power/resume" then attempts to resume from the first device mapper device.
Note that it is important to set the device in /sys/power/resume, regardless if resuming or not, otherwise later suspend will fail.
If resume starts, script execution terminates here.

Otherwise we just remove the encrypted swap device and leave it to the mini system on /dev/hda1 to set the whole crypto up (it is up to you to modify this to your taste).

What then follows is the well known process to change the root file system and continue booting from there. I prefer to unmount the initrd prior to continue booting but it is up to you to modify this.