Encrypted GNU+Linux with Coreboot and Grub

I will be using Parabola GNU+Linux, a fork of Arch, for this example. Most of it will apply to other distros.

Before You Start

compiling coreboot

grub options

These are the grub modules needed for encryption:

- crypto
- cryptodisk
- luks - if you are used luks encryption.
- procfs -
- archelp
- lvm - if you are using lvm.
- pbkdf2 - The Password-Based Key Derivation Function 2 module. Used for key stretching
- gcry_rijndael - or…
- gcry_sha512 - or whatever hash you used during cryptsetup

Booting live usb distributions using iso/syslinux:

- nativedisk
- squash4
- syslinuxcfg
- loopback

flashing coreboot

Some computers will allow you to flash the rom internally first time, others will be write protected so the first time will have to be external. Either way, it is recommended to get an external flasher incase you make a mistake.

launching distro live usb from grub

cryptsetup

cryptsetup is the commandline tool used to access the cryptographic features of the linux kernel.

After this you would install your system onto that partiton.

grub configuration

Before flashing a rom with this configuration, you can test these commands in the grub minimal commandline first.

You have to manually load the luks and lvm modules 'insmod luks'. It appears to load all it's requirements.

---

https://wiki.openwrt.org/doc/networking/network.interfaces

https://wiki.openwrt.org/doc/uci/network/switch

ip-link man page.