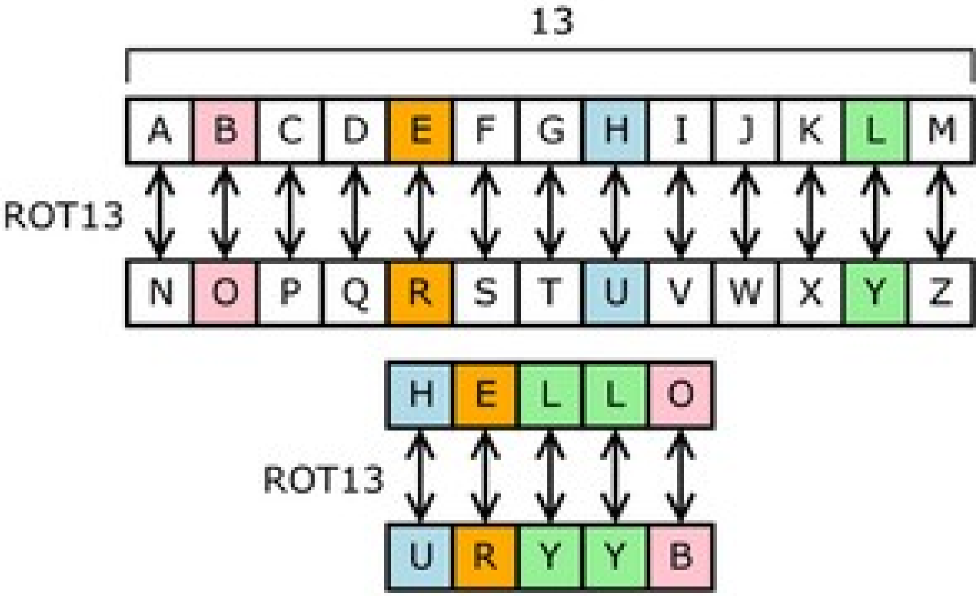
Linux and Cryptography

Dr Gareth Owen
University of Portsmouth

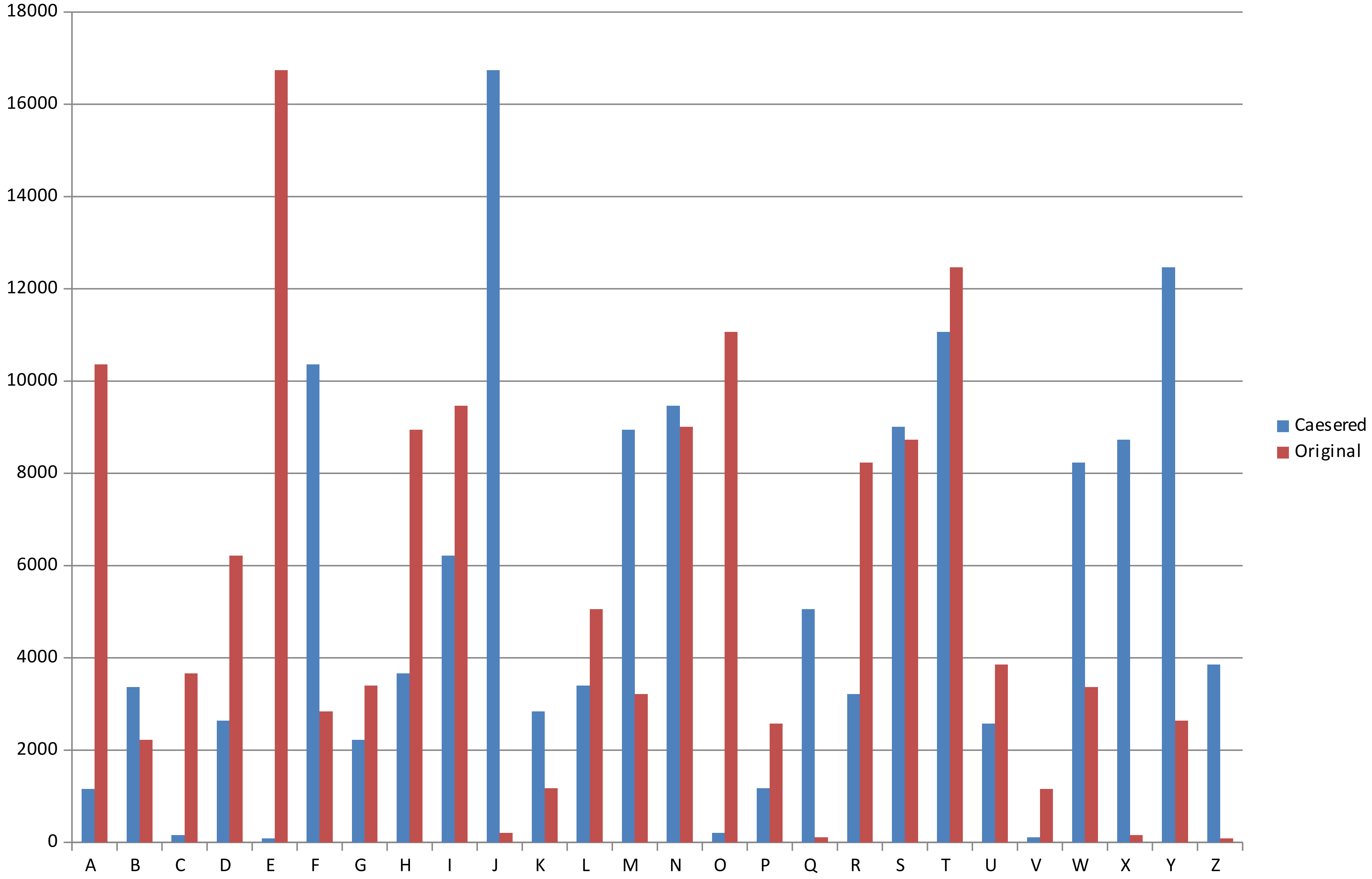
What is Cryptography?

ROT-13

- ROT-13 is a type of encryption used on newsgroups.
  - Caeser cipher with a shift of 13.
  - What properties does ROT13 have?



Dickens book Caesered



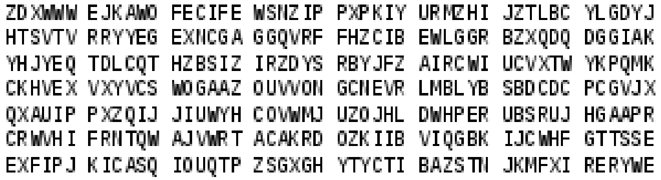
Substitution ciphers

- An extension of a caeser cipher
- Instead of shifting X number of letters through the alphabet, we have an alphabet which the original maps to.
- Eg:
- `ABCDEFGHIJKLMNOPQRSTUVWXYZ` ↕
- `QWERTYUIOPASDFGHJKLZXCVBNM`

XOR Cipher

- XOR encryption
-
- I AM A LUMBERJACK
- 49 20 41 4d 20 41 20 4c 55 4d 42 45 52 4a 41 43 4b
- PASSWDPASSWDPASSW
- 50 41 53 53 57 44 50 41 53 53 57 44 50 41 53 53 57
- Ciphertext:
  19 61 12 1e 77 05 70 0d 06 1e 15 01 02 0b 12 10 1c
- Numbers are HEX representation of the text - as the cipher text is not ASCII printable

The one-time pad



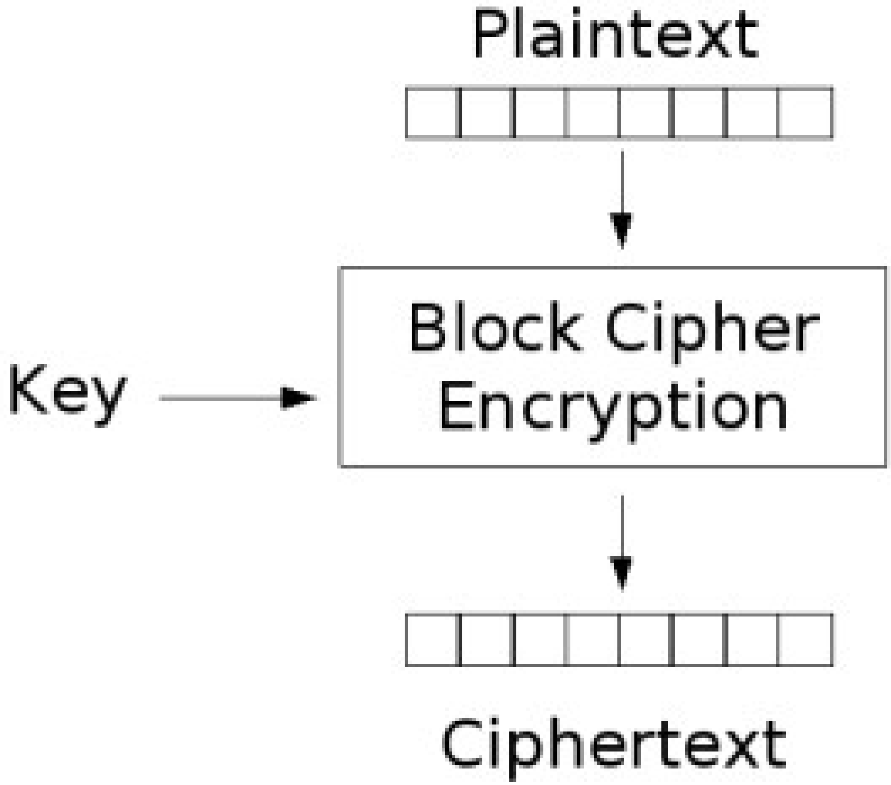
```
ZDXWWW EJKAWO FECIFE WSNZIP PXPKIY URMZHI JZTLBC YLGDYJ
HTSVTV RRYYEG EXNCGA GGQVRF FHZCIB EWLGGR BZXQDQ DGGIAK
YHJYEQ TDLCQT HZBSIZ IRZDYS RBYJFZ AIRCWI UCVXTW YKPQMK
CKHVEX VXYVCS WOGAAZ OUVVON GCNEVR LMBLYB SBDCDC PCGVJX
QXAUIP PXZQIJ JIUWYH COVWMJ UZOJHL DWHPER UBSRUJ HGAAPR
CRWVHI FRNTQW AJVWRT ACAKRD OZKIIB VIQGBK IJCWHF GTTSSE
EXFIPJ KICASQ IOUQTP ZSGXGH YTYCTI BAZSTN JKMFXI RERYWE
```

- First described in 1882
- Germans used it in 1923 – issuing code books
- Security proved by Claude Shannon in the 1940s.

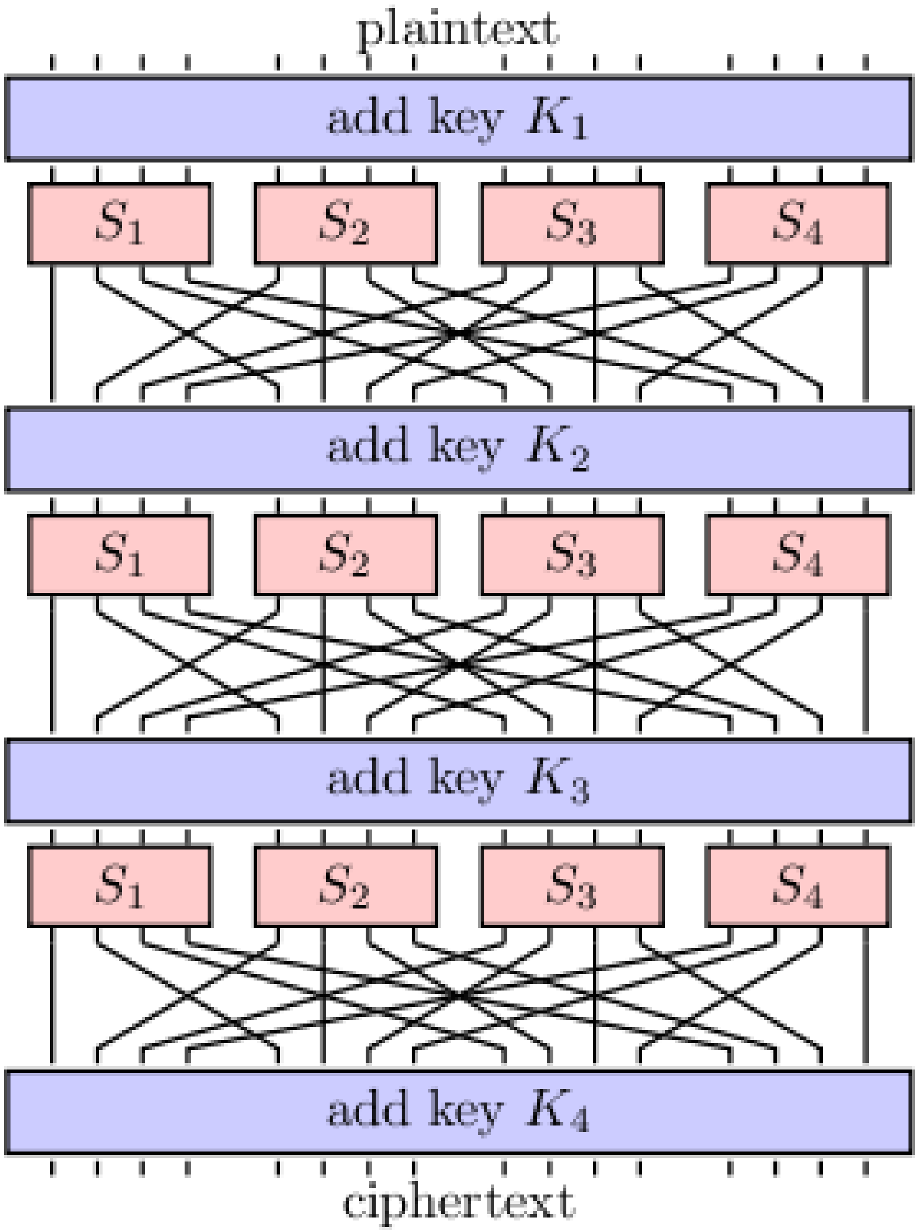
- Basic principle: Key as long as message, use XOR cipher or equivalent.
- Is "information-theoretically secure"
  - Ciphertext provides NO information about plain-text

Modern day block ciphers

- The output looks *like* random
- A single bit change in the input changes on average half the output bits.



Substitution Permutation Networks



Just how big does a key need to be? (symmetric cipher)

- Uses really big numbers
  - 1 in 2^{61} odds of winning the lotto and being hit by lightning on the same day
  - 2^{92} atoms in the average human body
  - 2^{128} possible keys in a 128-bit key
  - 2^{170} atoms in the planet
  - 2^{190} atoms in the sun
  - 2^{233} atoms in the galaxy
  - 2^{256} possible keys in a 256-bit key
- Note, non-elliptic curve asymmetric ciphers need >1024 bits which is roughly equivalent to 80-bits on a symmetric cipher. Elliptic curve ciphers need twice that of a symmetric cipher e.g. >= 256bits.

Thermodynamic Limitations*

- Physics: To set or clear a bit requires no less than kT
  - k is the Boltzman constant ($1.38*10^{-16}$ erg/ºK)
  - T is the absolute temperature of the system

- Assuming T = 3.2ºK (ambient temperature of universe)
  - kT = $4.4*10^{-16}$ ergs

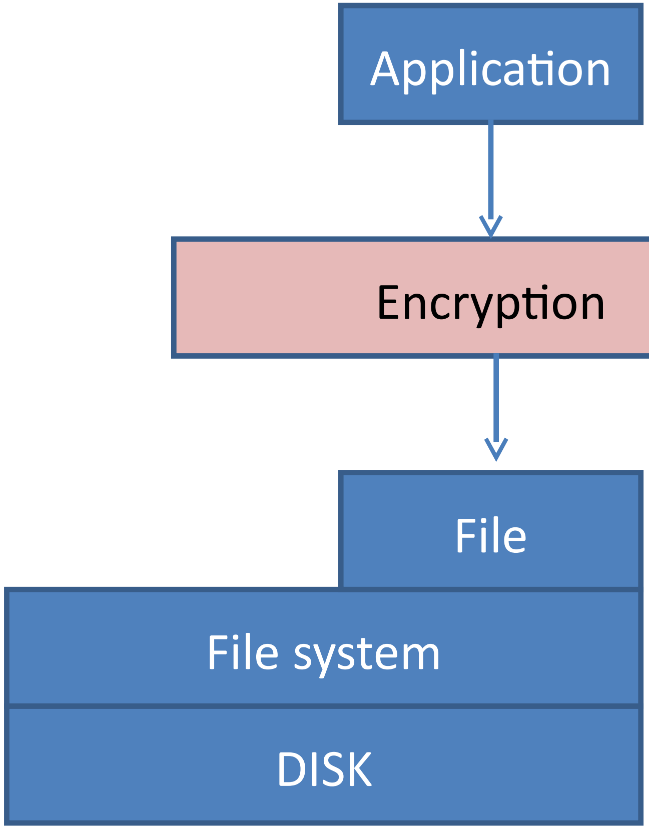
- Annual energy output of the sun $1.21*10^{41}$ ergs
  - Enough to cycle through a 187-bit counter

- Build a Dyson sphere around the sun and collect all energy for 32 year, we could
  - Enough to cycle through a 192-bit counter.

- Supernova produces in the neighborhood of 10^{51} ergs
  - Enough to cycle through a 219-bit counter

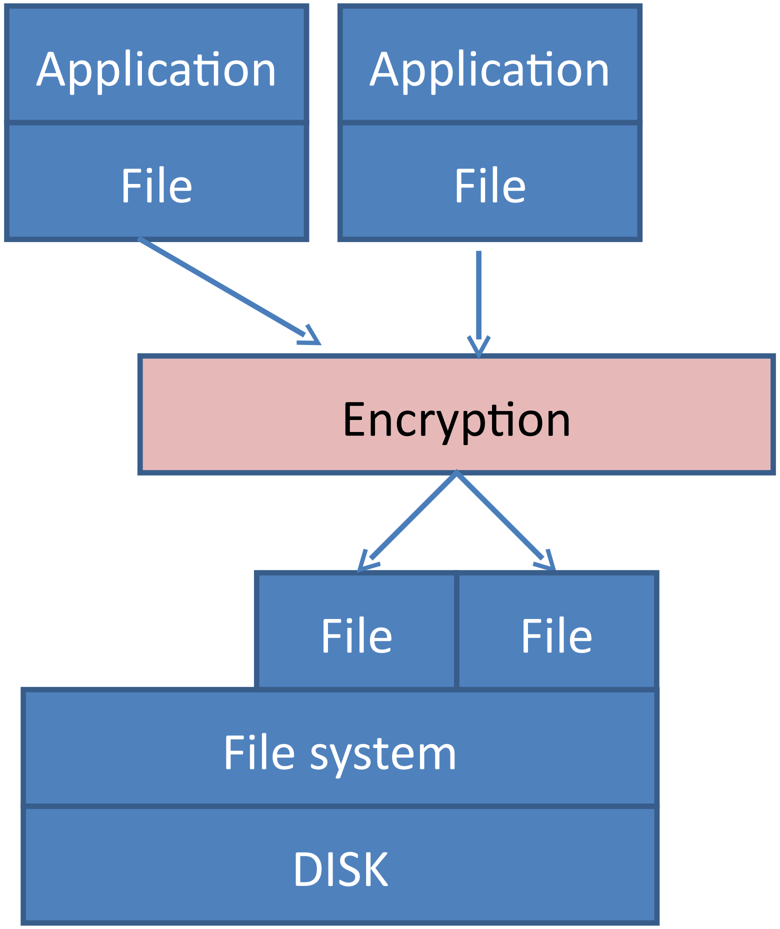
*From Applied Cryptography

File encryption



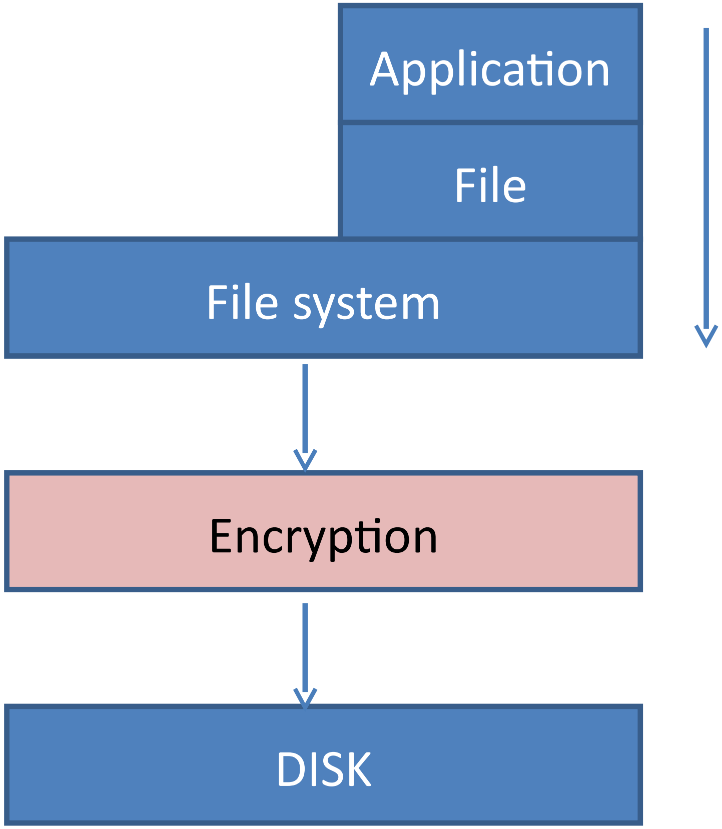
- File Encryption
- Examples
  - GPG
  - Zip
  - Word
- Advantages
- Limitations

Filesystem-level encryption



- Stacked file system
- Examples
  - EncFS
  - eCryptFS

Types of encryption



- Block Device Encryption
- Examples
  - TrueCrypt (tcplay)
  - BitLocker (disk)
  - dm-crypt
  - LUKS

Is encryption enough?

- In the UK, if you are arrested the Police may invoke S49 of the Regulation of Investigatory Powers Act:
  - You must surrender your password or face up to two years in jail.
- Solution: Plausibly deniably encryption
  - Recall: encrypted data looks like random (provided no header is added).
  - You can simply say that your disk is not encrypted and that you have erased it.
  - Not very plausible though is it? Although technically cannot prove either way.
  - What if your captor is an evil dictator?

What about e-mail encryption?

- We use asymmetric cryptography.
- You have two encryption keys
  - Public key which you give to everyone
  - A private key which you keep to yourself (and keep it secure).
- If you encrypt with one key, you need the corresponding key to decrypt.
- Public key is published to a key server where people can find it.
- Two choices: PGP (GPG is the tool) or S/MIME.

Current landscape

- Properly implemented cryptography is secure.
- NIST standardise many ciphers and the NSA successfully introduced a backdoor in 2006 to a random number generator… and then paid a major vendor to use it.
- Security Agencies are largely targetting end-points rather than trying to break the crypto – e.g. exfiltrate the keys.
- Encryption if it becomes widespread will mean going dark for the police and is currently thwarting investigations. RIPA s49 is their only limited defence.
- Known sanctioned backdooring or key escrow on a wide scale is not going to happen (I think).
  - Clipper chip in the 90s is a famous example which failed.
  - Yes, the does mean David Cameron is deluded.

Conclusions

- Rule #1 of cryptography is don't implement your own cryptography.
- Rule #2: Distrust priopprietry encryption.
- Recommendations for disk encryption:
  - Easy: LUKS
  - Plausibly Deniability: Truecrypt/tc-play
- Recommendations for e-mail encryption:
  - GPG – plugins available for most e-mail clients.
  - Web-mail is more difficult although there are some browser plugins.
- Recommendations for not getting caught:
  - Don't do anything wrong.

Having a go

- Create a disk image file:
- `dd if=/dev/zero of=diskimage bs=1024 count=50000`
- `cryptsetup luksFormat ./diskimage`
- `cryptsetup luksOpen ./diskimage dimg`
- Now look in /dev/mapper, you should see a file which represents your disk decrypted (dimg)
- Let's format it:
  - `mkfs.ext4 /dev/mapper/dimg`
- Now you can mount it:
  - `mount /dev/mapper/dimg /mnt`
- Any files you put in /mnt will be encrypted and stored in diskimage
- When done, umount then cryptsetup luksClose