A steganographic block store across several media files

- Final year project at the University of Cambridge
- Supervisor – Jean Martina

Overview

- Purpose - protection of privacy
- Stops people even knowing you have data
- Combines steganography and cryptography
- Creates a single store from several files
  - More media allows more data to be hidden
- No suspicious artefacts
  - E.g. No files which are obviously encrypted
  - E.g. No need for randomised hard drive

Steganography

- Hiding information
  - For the sake of this project, data in media files
- Provides a reason for storing random noise
  - Randomised hard disk is very suspicious
- Must be careful to preserve cover file
  - And, if possible, statistical properties of its bits
- Each media file presented as a block storage interface to its hidden data
- Any media with random noise can be used

Encryption

- Protecting information
- Good encryption will make data look random
  - Side effect of diffusion and confusion properties
- Random-looking data replacing random noise
- Infeasible to search all hidden channels for data

Splitting Across Media Files

- To make use of an entire directory
- Cover file contains start and end address
  - E.g. A file might contain hidden data from byte 5 to byte 505, so this is recorded
- Accessing the store – first make a list of which data is stored in which files

Redundancy

- Want ability to lose files
  - Some files might get corrupted
  - Overt attachment to files could be suspicious
- Split cover-files into groups
- Then use a RAID scheme on the groups
  - Treat each cover-file group like a separate HDD
- Try to make each group of similar size
  - In terms of storable hidden data
  - Total size is limited by size of smallest group

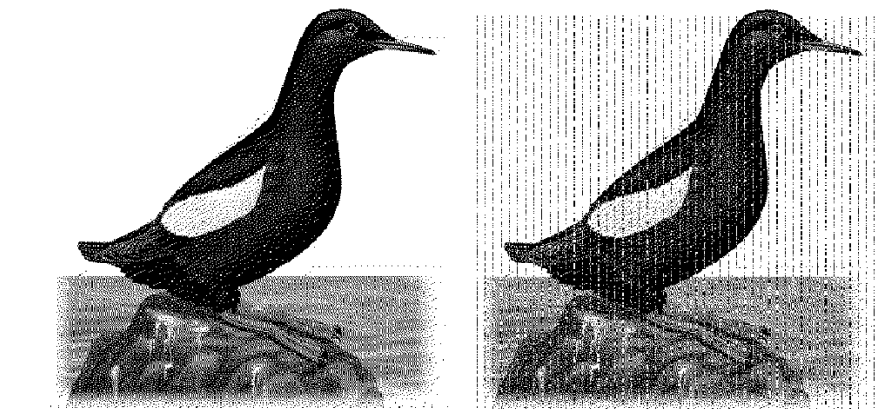
Encryption

- Uses a block cipher in counter mode
  - Nonce stored in plaintext in each file
  - Implicit counter increases through each file
  - Read/write to the middle of a file takes O(1) time
- Size of counter limits total size of store
- Safe, assuming nonces are never repeated
  - Places another limit on size of store
- Trade off between nonce and counter size
  - Because (nonce|counter) is fixed length

Privacy Provided

- Allows data to be covertly stored in media
- Computationally infeasible to detect data
  - Too much media in the world, plus encryption
  - Hence also infeasible to read data
- Stoppable by limiting transfer of media
- No suspicious activity required
- However – must still be careful with media

Example



Performance

- Average speed of ~25,000 bytes per second
  - A total of 1000 media files used
  - Divided into 3 groups for triple redundancy
  - Able to write 500,000 bytes in about 20s
    - Includes the necessary redundant writing
- Reading is about twice as fast

Limitations

- Need to keep internal system state secret
- Need to keep original media secret
- Attacker must not be able to see that you are changing the files
- Assumes that at least some files are left uncorrupted