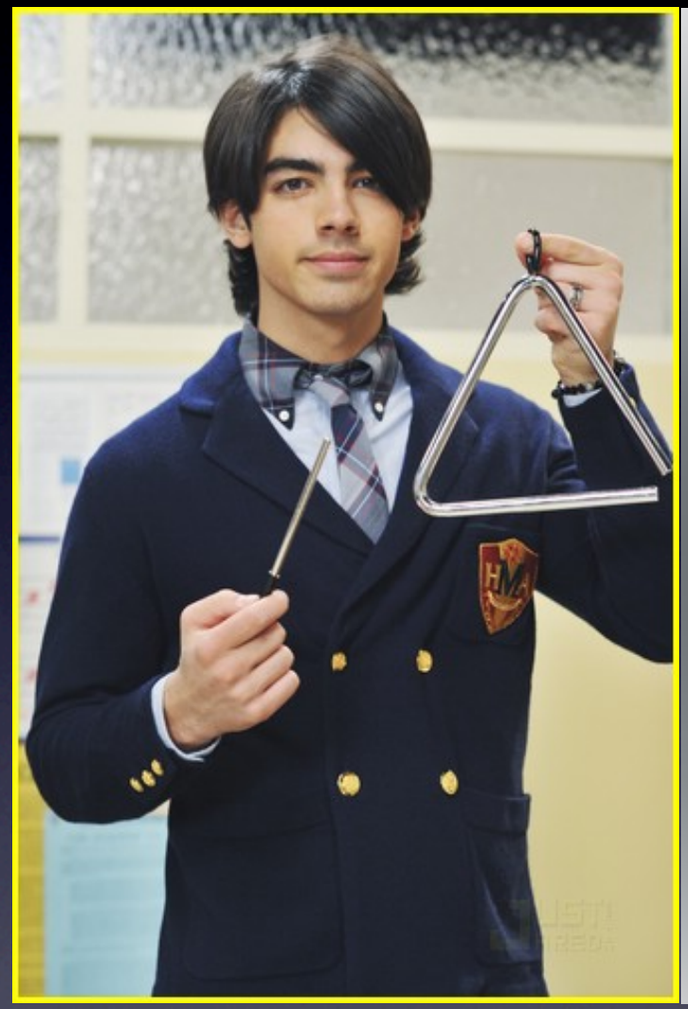
Practical Data Protection

Alistair Crooks
agc@netbsd.org

100% Dingbats free



Set up for later

VLC File Edit Playback Audio Video Window Help

Tx: 0B/s Rx: 0B/s U: 2728MB F: 1120MB 22% 18% (100%) Sat 22:44:04

3. Shell

```
[7:31:35] agc@amd64-vm2 ~/ebc2011/embarassing [1905] >
```

VLC File Edit Playback Audio Video Window Help Tx: 0B/s U: 2693MB Rx: 0B/s F: 1155MB 18% 19% (59%) Sun 4:30:39

3. Shell

```
[11:22:46] agc@amd64-vm2 ~/ebc2011/secrets [1995] >
```

media player

Data Protection

- Modification detection
- Error and erasure correction
- Protection from unauthorised viewing
- Redundancy, shares and thresholds

Modifications

- Checksums
  - lightweight
  - in widespread use
  - still in use in iSCSI and SCTP

CRC

- As a library
- scripting language access
- command line program is easy

Hash64

- Combine two 32bit hashes
- Result is 64-bits wide
- Much more resistant to collisions
- Can be extended to combine digests

HMAC

- Hashing Message Authentication Code
  - uses a shared key
  - uses underlying digest
  - result is more secure than digest

Hashtrees

- Digests tell us data has been changed
- Do not tell us which part has changed
- Re-transmit or recover - be smart

Lamport signatures

- One-time keys
  - can be mitigated with hashtrees
- Alternative to RSA/DSA/EC
- Key propagation?

Error Correction

- Error - octet modified, location known
- Erasure - octet modified, location unknown

Reed Solomon

- Add parity bytes to protect data
- (32, 28) Reed-Solomon coding
  - 28 protected bytes
  - 4 parity bytes

Circa

- CD-style protection for data
- Modified to fit in a data block
- 2348 bytes to 3136 bytes
- Size based on expanded/protected size

1. 4-byte 1 frame delay

- Every other 4 bytes is delayed 1 frame
- (a C3 frame is 24 bytes)
- to protect against a linear scratch

2 Intra-frame scramble

- Within a C3 frame, the bytes are scrambled
- (a C3 frame is 24 bytes)
- to mitigate proximity errors

3 Q-parity

- Q parity - C2 frame is created from C3
  - (a C2 frame is 28 bytes)
    - apply a Reed-Solomon (28,24) encoding
    - inserted at half-way point of C3 frame

4. Dispersal

- each byte is striped across sector
  - by offset from start of C2 frame
    - byte b in C2 frame f becomes
    - byte b in C2 frame f + b

5. P-Parity

- Reed Solomon (32,28) encoding
- creates 32-byte C1 frame

6. Delay 2

- even bytes delayed 1 frame in output
  - differs from ECMA standard CDs
    - CDs delay by sectors here
  - will add as an extension at later date

CD encoding summary

- 2348 input bytes
- 3136 output bytes
- 4/3 expansion, due to 2 Reed-Solomon

How effective?

- On a 2348 byte sector
  - Random: 103 bytes corrupted
  - Simple Row: 20 bytes corrupted
  - Column: 135 bytes corrupted

VLC File Edit Playback Audio Video Window Help Tx: 0B/s Rx: 0B/s U: 2740MB F: 1108MB 18% 17% (100%) Sat 23:55:20

3. Shell

```
[8:42:50] agc@amd64-vm2 ...local/src/circa-test [1965] >
```

Protection from Viewing

Encryption & Decryption

- Netpgp
- OpenPGP/GPG
- SSH
- OpenSSL Cert

Redundancy, shares & thresholds

- Where one copy just isn't enough
- Different protection for different data
- Thresholds
  - Requisite number of shares

SSSS

- size - same as original
- protection from viewing
- threshold/shares

VLC File Edit Playback Audio Video Window Help

3. Shell

```
[19:56:58] agc@amd64-vm2 ~/ebc2011/ssss [1691] >
```

ida.mov

Rabin's IDA

- threshold/shares * size
  - original 526 bytes, 3/10 threshold
    - each share (of 10) is 192 bytes
- protection from viewing
- threshold/shares at split time

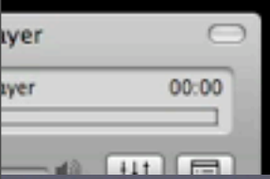
VLC File Edit Playback Audio Video Window Help

Tx: 0B/s U: 2622MB
Rx: 0B/s F: 1226MB

20% 18% (97%) Fri 3:43:30

3. Shell

```
[3:05:31] agc@amd64-vm2 ~/ebc2011/ida [1611] >
```



Mat - overview

- archiving tool
  - NOT tar/pax/cpio/zip compatible
- embedded sha256 checksum
- handles, links, symbolic links

Bag

- mat
- digital signature
- xz/liblzma compression
- circa

Enhancements

- Encrypt pass phrase to actors
- Use digital watermark during recovery
- Add encryption to user's key

Embarassing

VLC File Edit Playback Audio Video Window Help Tx: 0B/s Rx: 0B/s U: 2714MB F: 1134MB 23% 22% (100%) Sat 22:54:43

3. Shell

```
[7:41:51] agc@amd64-vm2 ~/ebc2011/embarassing [1924] >
```

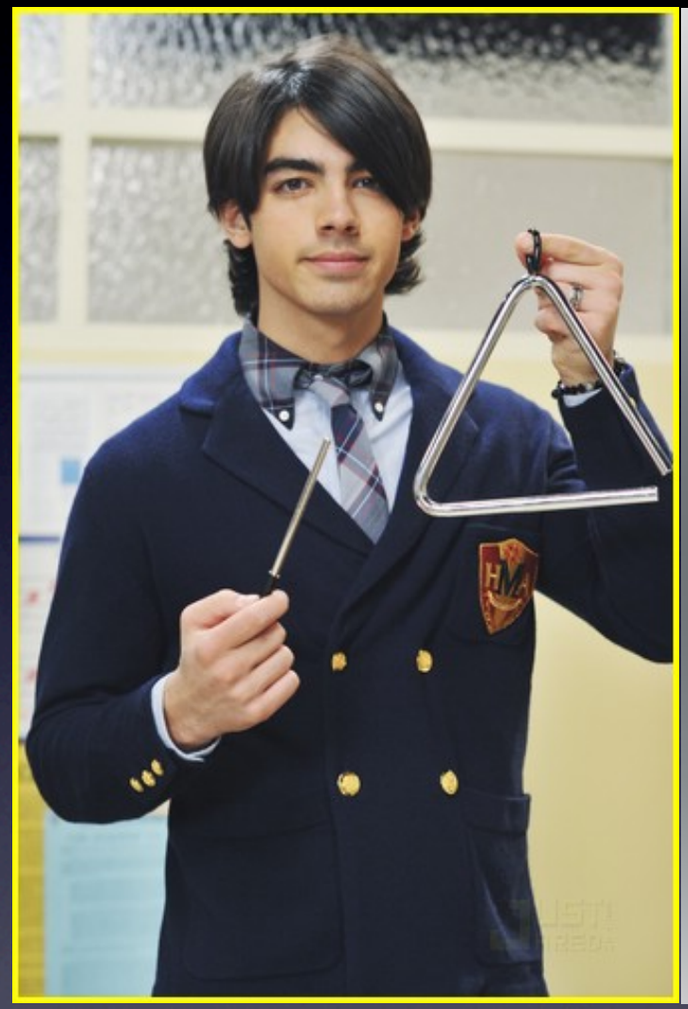
Shared Key Distribution

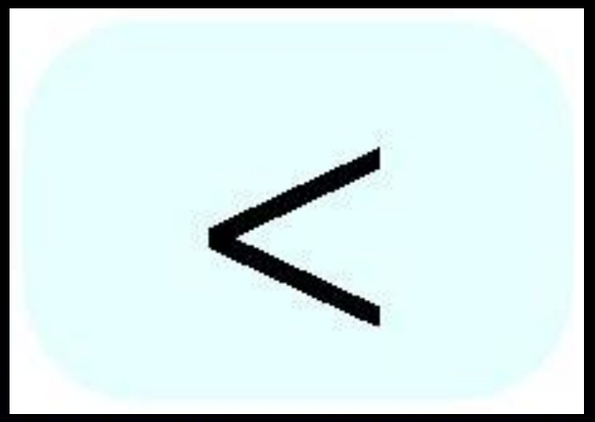
- Generate a one-time key
  - passphrase?
- Encrypt data to that key
- SSSS the secret key
- Distribute shares to actors

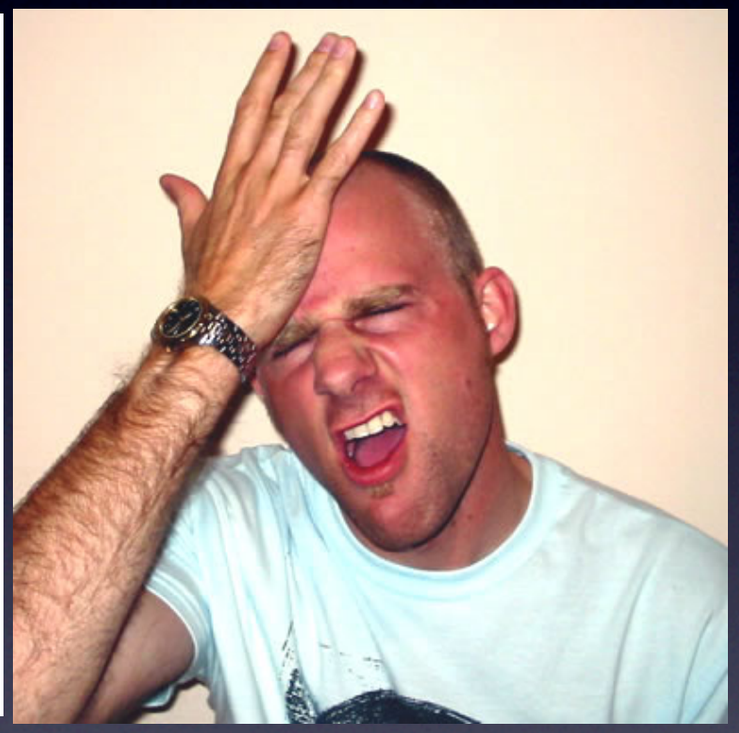
Shared Key Recovery

- Receive <threshold> shares from actors
- SSSS join the provided shares to get key
- Decrypt data with that key
  - Passphrase?
- Recovered data

100% Dingbats free







"Those who cannot remember the past are condemned to fulfill it"

-- George Santyana

Lessons Learned

- Checksum equality can cause inefficiency
- Use basic building blocks to build
- Protection and resilience are effective
- Thresholds useful

Secrets

VLC File Edit Playback Audio Video Window Help Tx: 0B/s U: 2698MB Rx: 0B/s F: 1149MB 20% 16% (65%) Sun 4:39:49

3. Shell

```
[11:31:21] agc@amd64-vm2 ~/ebc2011/secrets [2014] >
```

media player

Uses

- Recover (sensitive?) archived data
- Shared secret recovery from peer machines
- Rubber-hose style data within data
- Distributed authentication keys

Similar work

- par2 - GPLv2
- tahoe-lafs - GPLv2, distributed file system
- ldpc, turbocodes, tornado codes (patents)
- visual cryptography - one-time pads
- homomorphic encryption

Lessons Learned

- Circa can add file protection
- Use building blocks
- Thresholds provide file resilience
- Sharedkey - efficient, useful encryption

Questions?

Thanks!

- Yahoo! for sponsoring my talk
- Alistair Mccoist
- Guy who wiped me out on Big Basin Way
- BSD community

Alistair Crooks

agc@NetBSD.org