Making sure data is lost.

Spook strength encryption of on-disk data.

Poul-Henning Kamp

The FreeBSD Project

<phk@FreeBSD.org>
”A line in the sand”

- Before operation ”Desert Shield/Storm”, Air Chief Marshal Patrick Hine briefed the British Prime Minister on the battle plan.
- After the meeting, his aide forgot to lock the car while shopping.
- A briefcase and a laptop computer were stolen from the car.
A line in the sand...

- The briefcase (with documents) were subsequently recovered.
- The laptop and the copy of the battle plan on its disk were not.
- ”We sat down and hoped...”
Not all cops and users are stupid

• Most OSS disk encryption software suffer from soggy analysis.

• Cgd (OpenBSD/NetBSD)
  – You cannot change your passphrase without reencrypting the entire disk (takes a day).
  – One key for all sectors.

• STEGFS (Linux)
  – User cannot prove compliance.
GEOM Based Disk Encryption.

- Protect ”cold disks” with strong crypto.
- Protect user with proof of destruction.
- Filesystem/Application independent.
- Architecture and byte-endian invariant.
- Practically Deployable.
- Developed under DARPA/SPAWAR contract N66001-01-C-8035 ("CBOSS"), as part of the DARPA CHATS research program.
”Cold disks?”

- A ”cold disk” is one for which the corresponding key-material is not available:
  - CD-rom or floppy in the mail.
  - Disks in a file-cabinet.
  - Disk in computer which is turned off.
  - Computer which has not ”attached” to protected partition on the disk.
A ”cold disk” is not:

• A laptop in suspend mode.
• A computer with a screen saver.
• A disk with a ”Post-It” with the password.
• A disk with the password ”password”
File System Independent.

• Actually: ”Transparent to application”.
• GBDE works at the disk level and the encrypted partition looks like any other disk partition to the system.
  – Swap, UFS1/2, iso9660, FAT, NTFS, Oracle, MySQL &c, &c.
• Trickier to implement good crypto.
• Easier to use.
Byte-endian/architecture invariant.

- Important for media portability.
- Extend lifetime of algorithm to future computers.
Practically Deployable

- If crypto is too cumbersome, people will bypass it, rather than use it.
  - "We have to get work done too..."
- Multiple parallel pass-phrases.
  - Master key schemes.
  - Backup keys.
  - Destructive keys [future feature].
- Changable pass-phrases.
"Protected, how long time?"

- If I could predict the future, I wouldn't write software, I'd be making millions being a meteorologist.

- Depends on:
  - Future hardware development.
  - Yet undiscovered weaknesses in algorithms.
  - How well the pass-phrase(s) were chosen.
  - How large the media is.
  - Who the enemy is, and how much they care.
Crypto principles

- Standard algorithms
  - AES, SHA2, MD5 (bit-blending only)
- Primary strength delivered by crypto
- Secondary strength from frustrations
  - Unpredictable on-disk locations
- No two-way leverage
  - Random one-time use sector keys
Symmetric / Asymmetric keys

- Two kinds of keys:
  - Symmetric keys.
  - Asymmetric keys (public-key crypto).
- GEOM uses symmetric keys.
- PGP uses asymmetric keys.
- 128 bit symmetric $\approx 2304$ bit asymmetric.
So how strong is GBDE?

- Breaking 128 bits opens a single sector.
  - If you know where the sector is.
- Breaking 256 bits will open the entire thing
  - If you try all sectors to find the lock sector.
  - If you try a lot of variant encodings.
- Provided you recognize that you found a hit in the first place (expensive!).
Pointless Comparison

• A normal cylinder door lock has approx 2 bits per pin and 6-8 pins \(\approx\) 12-16 bits.

• (computer-)key to (door-)key conversion:
  – 128 bit \(\approx\) 20cm / 4” of door-key
  – 256 bit \(\approx\) 40cm / 8” of door-key
"What does Bruce Schneier say?"

- Personal affairs: 128 bit.
- Diplomatic embarrassment: >128 bit.
- U.S. Census data: >128 bit.
Summary

• GBDE protects data with:
  - At least $O(2^{128})$ work per sector.
  - At least $O(2^{256})$ work per disk.

• Reviewers agree so far that:
  - GBDE will not be broken, unless AES is significantly broken.
  - Far more productive to find the passphrase.
About that pass-phrase...

• This is a 64 bit pass-phrase:

Blow, winds, and crack your cheeks! rage! blow!  
You cataracts and hurricanoes, spout  
Till you have drench'd our steeples, drown'd the cocks!  
You sulphurous and thought-executing fires,  
Vaunt-couriers to oak-cleaving thunderbolts,  
Singe my white head! And thou, all-shaking thunder,  
Smite flat the thick rotundity o' the world!  
Crack nature's moulds, and germens spill at once,  
That make ingrateful man!
Storing pass-phrases.

• A good pass-phrase must be long, subtle and not a direct quote from Shakespeare.

• People cannot remember it.

• GBDE can take pass-phrase from anywhere
  – Keyboard, USB-key, Chip-cards, &c &c.

• Pass-phrase need not be text:
  – SHA2/512 hashing of passphrase allows it to be any bit sequence.
Augment your pass-phrase.

- Make your passphrase consist of two parts:
  - The stuff you type in from the keyboard
  - 1-8 kbyte of random bits stored on USB key.
- ”Something you know + something you have” principle.
- Other ideas:
  - 1wire buttons
  - Smart cards.
Getting rid of data, fast!

- Sometimes you want to destroy data fast:
  - Students taking over the embassy in Tehran.
  - State police raiding human rights offices.
  - RIAA raiding college dorms.
  - Wife asking ”What takes up all those 40 Gigabytes on our hard disk ?”.

GBDE as vault dynamite.

- The user can destroy all lock sectors.
  - 2048 + 128 bit master key is erased.
  - Attacking disk now requires $O(384)$ work.
  - $384 \gg 256$

- Positive feedback that lock is destroyed.

- But data can still be recovered by restoring encrypted lock sector from backup.
Uses of four lock sectors

• Media initialized by IT department:
  – Initialize locksector #1 with master pass-phrase.
  – Put backup copy of locksector #1 in safe.
  – Initialize locksector #2 with user pass-phrase.
  – Erase lock sector #1 from disk.

• User can change his own pass-phrase.

• IT dept can recover when:
  – user forgets pass-phrase.
  – user destroys lock sectors.
How to initialize GBDE:

• Put "GEOM_BDE" option in your kernel.
  – or kldload module "geom_bde"
• # gbde init /dev/ad0e
• Enter new passphrase: __________
• Reenter new passphrase: __________
How to create filesystem on GBDE:

• # gbde attach ad0e
• Enter passphrase: ______
• # dd if=/dev/random of=/dev/ad0e.bde bs=64k
  – Fills disk with encrypted random bits.
• # newfs /dev/ad0e.bde
• # gbde detach ad0e
How to use GBDE:

• # gbde attach ad0e
• Enter passphrase: ________
• # fsck -o /dev/ad0e.bde
• # mount /dev/ad0e.bde /secret
• (do work)
• # umount /secret
• # gbde detach ad0e
HW assist crypto

• I have unfinished code for HW assisted crypto using OpenCrypto framework.
• Some outstanding issues to be fixed.
• Works with the Soekris VPN14x1
  – Hifn based miniPCI or PCI card.
  – Approx $100.
• Not tested with other hardware.
Firewire is evil!

- If your computer has a firewire port a screen saver gives you no security.
- Firewire allows all of RAM to be accessed by any device which plugs into your firewire port.
- Solution:
  - Glue and toothpicks.
Availability

- GBDE is in FreeBSD-5.0 and later.
- The algorithm can easily be ported to any other operating system.
  - You do not need to take all of GEOM along.
- Paper & slides about GBDE:
  - http://phk.freebsd.dk/pubs/
Conclusion:

- GBDE will encrypt your data with at least 128 bits symmetric key, and your passphrase will be the weakest link.
- Very flexible keying scheme can be used to deploy it in real-world scenarios.
- **DON'T FORGET YOUR PASS-PHRASE!!!**
  - I can't help you get your data back.