Complete Delete vs. Time Machine Computing

Simson L. Garfinkel
Center for Research on Computation and Society
Division of Engineering and Applied Sciences
Harvard University
simsong@acm.org

ABSTRACT
Users are increasingly demanding two contradictory system properties – the ability to absolutely, positively erase information so that it cannot be recovered, and the ability to recover information that was inadvertently or intentionally altered or deleted. Storage system designers now need to resolve the tension between *complete delete* and *time machine computing*.

Categories and Subject Descriptors
E.5 [**Data**]: Files

General Terms
Reliability, Security, Human Factors

Keywords
Complete Delete, Time Machine Computing, File Vault, Secure Empty Trash, MacOS

1. INTRODUCTION
Two exciting storage security products will reach the market in 2007. One will allow users to instantly delete an entire hard drive's worth of information so that it can never be recovered. The second will allow computer users to recover information days, weeks or even months after it has been changed or deleted, whether by accident or on purpose.

2. THE TENSION
Clearly, the goals of these two systems are in conflict. What's more, the way that technologists decide to resolve these conflicts will have deep implications for personal privacy, corporate security, law enforcement and even national security. Should computer systems provide their users with the ability to absolutely delete information, or should some kind of information trace be preserved if at all possible? Should users have the ability to selectively completely delete information—akin to removing pages from a diary or accounting ledger without leaving a trace? Or should users only have the ability to take a computer and "reset to installation," wiping out all information at the same time and leaving an affirmative record that a wipe operation has taken place?

The increase in storage capacity afforded by modern hard drives combined with increasingly powerful trusted systems and pervasive connectivity means that we can build systems that implement any policy that we can clearly articulate. Indeed, commercial variations on all of these themes are being introduced into the marketplace in 2007. How we choose to resolve these conflicts will ultimately be an exercise in both engineering and policy making.

2.1 Complete Delete
In January 2003, Shelat and I reported that roughly one third of 158 hard drives purchased on the secondary market between November 2000 and August 2002 contained confidential or highly sensitive information that should never have been released.[4] For example, one drive had been used in an ATM machine and still contained customer financial information. Another drive contained more than 3,700 credit card numbers from a terminal that had been used to submit charges from a supermarket to a bank.

Shelat and I hypothesized that many of the people who had left confidential data on the drives in our study had attempted to delete the information but had failed in their attempts. In many cases individuals had explicitly deleted the files containing sensitive information, apparently unaware that deleted files are not overwritten until the space is needed for other purposes. In other cases the drive's previous owners had used the Windows FORMAT command to wipe the hard drive, not realizing that the command doesn't actually overwrite file data. We were able to recover the data left behind using forensic tools, but the previous owners, having only the operating system tools at their disposal, would have reasonably thought that they had removed the confidential information. Our suppositions were confirmed by follow-up interviews.[3]

For years there have been third-party utilities for selectively overwriting individual files or even entire hard drives. But these utilities have two big drawbacks: because they are not included with the operating system, many users don't know about them. And because these utilities rely upon overwriting to erase information, they can be quite slow. For example, Apple added a "Secure Empty Trash" feature to its MacOS operating system following the publication of the 2003 study to give users a reliable means to remove confidential information from their hard drives. Secure Empty Trash uses a seven-pass overwrite for each file being deleted to assure that the information cannot be recovered by any means, and is very slow as a result.

But in principle there is no reason that this kind of "complete delete" functionality needs to be slow. Secure Empty Trash could do its overwriting in the background. What's more, for disk drives manufactured after 2001, a single overwriting pass is now generally regarded to be sufficient.[6] And by utilizing cryptography, sanitization can be made virtually instantaneous.

2.2 Cryptographic Erasing
In 1996 Boneh et al. proposed a tape backup system "[that] applies cryptography in a new way… to erase information rather than protect it." [1] Boneh's scheme encrypted files as they were written to a backup tape, storing the key for each backup of each file in a master key file. At a later point in time a specific version

of a backed up file could be "revoked" by removing the key corresponding to that file and version. Alas, Boneh's system did not provide for total data destruction because "the removed [key] can still be found in the backup version of the key-file."

Boneh's system can be improved by implementing it and the required key management directly inside a storage device. For example, Garfinkel and Shelat proposed equipping disk drives with "a cryptographic subsystem that automatically encrypts every disk block when the block is written, and decrypts the block when it is read back. Users could then render the drive's contents unintelligible by securely erasing the key."[4]

This spring Seagate will introduce this approach exactly in the company's Moments 5400 FDE.2 disk drive. The drive features Seagate's new DriveTrust technology, an integrated encryption module that provides for full disk encryption, drive pairing (locking a drive to a specific host), cryptographically hidden partitions, and secure erase and disposal. "If the encryption key is changed or eliminated, all of the data is instantly rendered inaccessible."[9] Seagate calls this technology "Crypto Erase."

(Decru incorporated a similar approach into its line of enterprise storage security appliances in late 2003, calling the approach "CryptoShred."[2] But because it was designed to be used in a data center, the Decru technology would not have prevented most of the data incidents that Garfinkel and Shelat uncovered.)

Technologies like Crypto Erase and Secure Empty Trash make it relatively easy for users to wipe files and media so that data cannot be recovered. Although it is commonly reported in the popular media that it's all but impossible to delete information from a computer in such a way that it can't be recovered by a trained forensic examiner, in practice this is no longer true.

2.3 Time Machine Computing

At the very same time that systems are incorporating better technology for permanently erasing information, they are also getting improved technology for recovering information that's been erased—either accidentally or intentionally.

Apple's Time Machine, schedule to be released with MacOS 10.5, is perhaps one of the best examples of easy-to-use file recovery technology. Time Machine automatically writes files that are changed to a chronologically-indexed database residing on an external hard drive. Users in the future can recover data that has been changed or deleted by clicking the Time Machine icon and then going "back in time" — that is, by searching chronically backwards through the archive until the desired information appears.

Although incremental backups have been used for decades and Rekimoto demonstrated time-machine computing in 1999[8], Apple's Time Machine is likely to stand out for several reasons:

1. By integrating with existing applications like Apple's Finder and Address Book, Time Machine lets users browse through their backups using graphical user interfaces that they have already mastered. The only new interface that needs to be mastered is the temporal browser, which appears to be very simple.
2. By utilizing the operating system's ability to report changes, Time Machine eliminates the need to continually scan the entire system for changes. This reduces the overhead of running the program which, consequentially, making it more likely that Time Machine will actually be run.
3. Unlike traditional backup systems that were designed to work with serial-access storage devices like tape, Time Machine is designed to work with high-capacity random-access storage devices as typified by external hard drives. Such drives are now cheaply available and have capacities in the hundreds of gigabytes

According to Apple, less than 25% of its users back up their computer in any way, and only 4% make ongoing backups—this, despite the fact that Apple's dot-Mac service includes an automated online backup system. Because it combines simplicity and comprehensiveness, Apple's Time Machine could prove to be quite popular and a model for future backup systems on other platforms.

2.4 Reconciling Complete Delete and Time Machine

What is the proper way for complete delete technologies like Apple's Secure Empty Trash and Crypto Erase to interact with pervasive backup technologies Apple's Time Machine? Since Time Machine hasn't shipped to customers, we don't know how Apple will address this real conflict between the desire to permanently delete information and the desire to recover information that is accidentally lost. Indeed, no matter how Apple ultimately addresses this question in MacOS 10.5, this is sure to be a question that is hotly debated in the coming years—and not just at Apple's headquarters in Cupertino, but among the users of all computer systems, and perhaps even by lawmakers in Washington and other national capitals.

If the user drags a file to the Trash Can and then chooses Secure Empty Trash, MacOS could simultaneously delete the file's backups from Time Machine. There have been several posts in MacOS user forums from users who say that this is the behavior that they expect. On the other hand, accidentally deleting files with Secure Empty Trash seems to be the very sort of mistake that Time Machine should protect against.

Even if Secure Empty Trash *should* delete the backup from Time Machine, this might not be possible. Time Machine relies upon an external hard drive to keep its backup. One of the advantages of this approach is that it makes disaster recovery a lot easier. If a laptop's hard drive crashes or the laptop is lost, Time Machine can reload the user's backup onto a computer with a newly installed copy of MacOS. But what should Secure Empty Trash do it if the external drive is not connected when the user invokes the command: should it warn the user that the backups will not be securely deleted, or should it remember the command and delete the backups when the drive is later attached?

One way around this conundrum would be to gimmick both commands so that Secure Empty Trash is disabled if Time Machine is operational. But this seems like the wrong approach as well.

Faced with this sort of quandary, many programmers would throw up their hands and give the choice to the user. Perhaps an alert box should appear: "You have chosen Secure Empty Trash, but many of these files are also present in your Time Machine backup. Do you wish to erase the Time Machine copies as well?" If the user chooses "yes," then the computer could insist that the Time Machine drive be plugged in so that it could be properly

scrubbed. Alternatively, if the files on Time Machine were protected with a backup system that supported CryptoShredding, the per-file encryption key could simply be erased.

The problem with all of these approaches is that the user who has chosen Secure Empty Trash is likely to be equally sanguine about deleting the Time Machine Backups—especially in those very times that the user is making a mistake. For example, the user might be deleting the wrong files. Or the user doing the deleting might not be authorized to do so—for example, my daughter might be deleting my tax returns because she needs more space for downloading movies, and she might have chosen "Secure Empty Trash" because she didn't want to leave a trace of what she had done. The user might even be attempting to hide illegal activity.

In the 1990s the US government proposed that industry adopt "CLIPPER Chip" which would have given consumers and businesses strong encryption, but give the US government a back door to the data. The proposal was rejected by businesses and consumers alike. However, if strong deletion technologies create problems for law enforcement, there may be similar calls to control the technology. One can imagine a strong delete system resource that only deletes information if the deletion action is logged with a centralized service, effectively allowing people to destroy evidence but not to hide the fact that evidence has been destroyed. A more invasive solution might not delete data at all, but merely re-encrypt the data using a key that was only available to law enforcement operating under the appropriate legal authority.

2.5 Delayed Unrecoverable Actions

One way to resolve some of the tension between perfect deletion and perfect retention is to retreat from immediacy and absolutes.

Norman observed in 1983 that simple confirmation boxes (e.g. Figure 1) for unrecoverable actions frequently fail to prevent error on the part of the user because the act of confirming the action is rapidly assimilated into the act that the box is intended to confirm. "the normal response to requests for confirmation is something like this: "Yes, yes, yes, yes. Oh dear!" [7]

As an alternative, Norman suggests a mechanism in which "the command can act as if it were actually executed, when in fact, it has only been deferred."[7] The computer executes the command at a later point in time, presumably after the user's attention switches focus, allowing the unconscious mind the opportunity to examine the action. We may call this kind of command a "delayed unrecoverable action." (Amazon.com's 1-Click Express Ordering System [5] is another example of a delayed unrecoverable action, in that 1-click orders can be changed or canceled after they are made but before they are shipped.)

Figure 1: Confirmation boxes such as this (also known as "swat boxes") frequently do not achieve their designer's goals of having the user consider the effects of an unrecoverable action, because clicking the "OK" confirmation becomes part of the action that the box is intended to confirm. An alternative approach is to let users initiate their actions but give them an opportunity to change their mind at a later point in time.

Delayed unrecoverable actions can be combined with cryptographic erasing in an interesting way. Instead of erasing the entire key, initiating a cryptographic erasure could instead erase one bit of the key every hour. The result is to make recovery of the deleted information computationally harder for each passing period of time. Within a few hours, the information could be speedily recovered. But after a day, it would take 4 hours of computer time to recover the cryptographic key (assuming that the computer could search through 1000 keys every second.) After two days it would require 8925 years of computer time to recover a key—completely within the realm of today's grid computers, but not a trivial undertaking. After three days, the data would not be recoverable for any practical purpose. An algorithm with a slower ramp to unrecoverability would be to randomly set one of the key's 128 bits to a 0 every hour.

3. CONCLUSION

Modern computer systems are simultaneously making it easier to delete information forever and making it easier to retain information after it has been accidentally or intentionally deleted. Interestingly, both of these capabilities will be deployed to consumers in Apple's MacOS 10.5 operating system, scheduled to be released later this year. What is not clear is how these two apparently irreconcilable features should interact with each other and with the user. We pose this as an open question for storage and usability experts alike.

4. REFERENCES

[1] Boneh, D. and Lipton, R., "A Revocable Backup System," Department of Computer Science, Princeton University, Princeton, NJ.

[2] Decru, Inc., "Decru Ships Decru DataFort T520 Security Appliances to Secure Data for Tape Backup," Decru Press Release, December 8, 2003.

[3] Garfinkel, S. "Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable," PhD Thesis, Massachusetts Institute of Technology, June 2005.

[4] Garfinkel, S., and Shelat, A., "Remembrance of Data Passed: A Study of Disk Sanitization Practices," *IEEE Security and Privacy*, January/February 2003.

[5] Hartman, P., Bezos, J., Kaphan, S., Spiegel, J., "Method and System for Placing A Purchase Order Via a Communications Network," US Patent 5,960,441, filed September 12, 1997, granted September 28, 1999.

[6] Kissel, R., Scholl, M., Skolochenko, S. and Li, X., "Guidelines for Media Sanitization," NIST Special Publication 800-88, September 2006.

[7] Norman, D. "Design rules based on analyses of human error." *Communications of the ACM*, 26(4), April 1983.

[8] Rekimoto, J., "Time-machine computing: a time-centric approach for the information environment," 12th ACM Symposium on User Interface Software and Technology, Asheville, NC, 1999, pp. 45-54.

[9] Seagate, "DriveTrust — FAQs," October 31, 2006.