The Good, the Bad and the Ugly of Consumer Cloud Storage

Wenjin Hu, Tao Yang, Jeanna N. Matthews
Clarkson University
{huwj,yangt, jnm}@clarkson.edu

Abstract

The promise of automatic data backup into the cloud is alluring. Off-site backup offers protection against a whole class of catastrophic risks (fire, flood, etc.) that on-site backup solutions cannot. Data can be backed up into the cloud automatically with little or no user involvement. Incremental backup software running detects the latest changes, encrypts the data, and sends it into the cloud. Files can be restored on demand and some services allow copies of files to be downloaded through a web interface to other machines , providing a form of file sharing. With costs dropping to ~$60-$100 per year for unlimited storage, it is not surprising that many home and small business users are signing up. In this paper, we evaluate four popular consumer cloud storage offerings – Mozy, Carbonite, Dropbox, and CrashPlan – to determine if they live up to the benefits users expect. We document wide variations in backup and restore performance, the type of data that is backed-up, no liability for data loss, and problems with data privacy. From our experiments, we derive a set of lessons and recommendations for consumer cloud storage that if followed more uniformly, could substantially improve the cloud storage experience for many consumers.

1. Introduction

Consumer cloud storage or online backup services are not new, but as with many aspects of the cloud computing landscape, they are becoming increasingly popular. There are many service providers to choose from and new service providers enter the market on a regular basis. Comparisons of available providers tend to focus primarily on price, ease of use, and the stability of the providing company. These are all very important factors, but it is harder to find comparisons that shed light on their architectural differences, provide solid data on their performance, and evaluate the guarantees, privacy and security they provide.

In this paper, we evaluate four popular consumer cloud storage offerings: Mozy, Carbonite, Dropbox, and CrashPlan. In Section 2, we present measurements of backup and restore times. We describe the set of tests we constructed to explain wide variations in the performance that we observed. Our results reveal interesting architectural differences in the services including how they react to network problems, whether they compress data prior to transfer, how they handle duplicate data, the amount of work they shift onto the client machine, and the degree to which they successfully overlap preprocessing of data with network transmission.

In Section 3, we highlight important variations in the features of each system such as what data is backed up by default and what types of data cannot be backed up at all. In Section 4, we evaluate the terms of service to reveal what guarantees service providers offer to end users. In Section 5, we discuss problems with private key management and potential privacy problems such as those introduced by content sharing between accounts.

Throughout the paper, we highlight two categories of lessons learned – warnings for consumers of cloud storage services and best practices for service providers. We conclude that there are substantial underlying differences between service providers that should be considered. Given the difficulty of moving from one service provider to another, it is especially important to provide consumers with solid technical data on which to base their choice. By highlighting the differences, we hope to shape the baseline set of features that consumers expect. Further, we hope to spark a discussion of what role the systems community can play in raising the standards for cloud storage services and possibly in shaping the legal rights users have with respect to the data they store in the cloud.

2. Back-up and Restore Performance

The most fundamental requirement for a cloud storage provider or online backup service is to backup data efficiently and allow for it to be restored on demand. Therefore, we begin by examining the backup and restore performance of four systems - Mozy [Mozy], Carbonite [Carbonite], CrashPlan [CrashPlan], and DropBox [DropBox].

We performed all of our experiments using a Pentium 4 3 GHz machine running Windows XP Professional Service Pack 3 with 512MB of memory, an 80 GB hard drive, and a Time Warner Cable network connection. By using an older machine and a residential Internet connection, we attempted to represent a typical home or small business environment. On this platform, we examined Mozy-1_16_4_0-9888, Carbonite 3.77 build 404, CrashPlan_2010-03-08_win, and DropBox0.7.110. When possible, we performed backup and restores at night when there was less contention for network bandwidth.

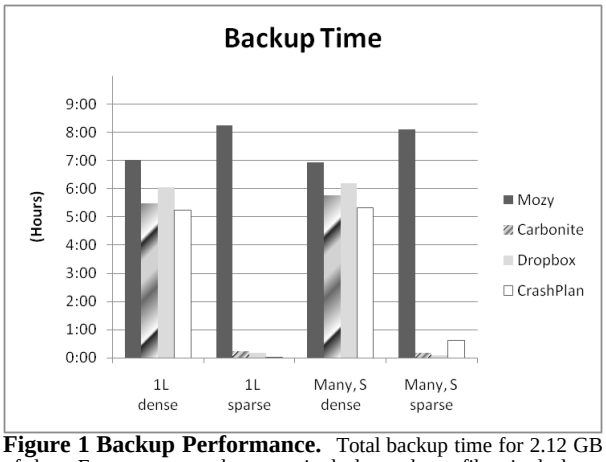

Figure 1 Backup Performance. Total backup time for 2.12 GB of data. Four cases are shown – single large dense file, single large sparse file, many small dense files and many small sparse files. The client machine is otherwise idle during the tests. Data contents are varied between experiments such that data has not previously been uploaded to the server.

2.1 Backup Performance

Our initial measurements were of the time to backup various 8GB files. We found wide variation in backup times from 10 minutes to 30 hours to still in-progress after 4 days. We saw substantial differences between the providers, but we also saw substantial differences for the same provider as we varied file names and file contents. In this section, we describe the set of tests we constructed to explain many of these variations.

The first mystery that we tackled was why some backups did not complete. We narrowed the problem down to files of 4GB or larger for Mozy. We pursued this issue with Mozy's tech support and they informed us that it is limitation of our network service provider rather than a limitation of the Mozy service. Indeed, Time Warner cable does appear to block long running TCP connections (e.g. when the data transferred approaches 4 GB in order to disrupt P2P sharing traffic). However, the other service providers were subjected to the same limitation, but they detect the network problem, establish a new TCP connection, and resume the backup from the point at which they were interrupted. Mozy, on the other hand, attempts to restart backup of the current file from the beginning. For large files, backups stay continually "in-progress" and never actually completes. Our first lesson is clear: *It is important to be resilient to network failures and to enable restart and incremental backup of large files.*

To avoid, this problem in the rest of our testing, we standardized on data sets of approximately 2GB – some consisted of a single large file and some consisted of many small files. Still, we saw wide variation in backup times from 3 minutes to over 8 hours.

Figure 1 isolates one factor in this variation. It shows backup performance in four specific cases. In the first case, we back up a single large file, the contents of which has been pre-compressed. We call this a dense file. In the second case, a single 2.12 GB file is also used, but in this case, the file is sparse and thus trivially compressible.[1]

The difference in backup performance for these two cases reveals that Carbonite, Dropbox, and Crashplan all compress data before transferring it to the remote server, while Mozy does not. For systems that do compression, the average backup time is 10 minutes for the sparse file, while Mozy takes over 8 hours. The backup times include any time for compression. There is little doubt that a brief period of additional CPU utilization is preferable to 8 hours of network overhead. This is a clear lesson for cloud service providers: *Cloud storage providers should compress data prior to transfer to a remote server.*

The remaining two cases also transfer 2.12 GB of data, but in the form of many smaller files – specifically 32 directories each with 16 4MB files. The contents of each file is different. In the third case, each of the 4 MB files are dense and in the fourth case, they are sparse and thus trivially compressible. In general, performance on many files is similar to the performance on one file. CrashPlan reliably shows a longer time for many sparse files than for a single sparse file.

Figure 2 reports the total amount of data transferred between client and server during each backup test as reported by BMExtreme Network Monitor [BMExtreme]. In many cases, the differences in total backup times illustrated in Figure 1 can be explained by the difference in the amount of data transferred over the network as shown in Figure 2.

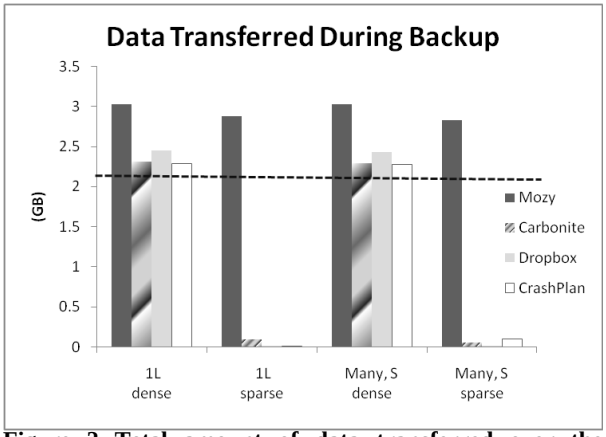

Figure 2 Total amount of data transferred over the network during backup. Four cases are shown – single large file dense, single large sparse file, many small dense files and many small sparse files. In all cases, the total data to be backed up is 2.12 GB as illustrated by the horizontal line at 2.12 GB.

1 The dense or pre-compressed files are created by first, compressing a large file with a series of four different compression methods: 7zip, zip, tar, gz and then cutting portions of the desired size from the pre-compressed file. The sparse or trivially compressible files are created by filling the entire file with zeros and then writing a short random non-zero string at the beginning of each 4K block.

Figure 2 reveals that in some cases, especially for Mozy, more data is transferred over the network than is contained in the original file. Where the amount of additional data transferred is substantial, this suggests that the client program is producing some type of metadata locally and then sending it to the server[2]. Interestingly, if we look at the ratio of data sent over the network during backup and during restore, we see that this same metadata is not returned to the client on restore. This suggests an off-loading of work from the server to the client. End users would clearly prefer that service providers do not off-load work onto their machines, but without clear data to highlight the difference they may not recognize the difference and that may give service providers an incentive to off-load more work where possible.

For the results shown in Figures 1 and 2, we took care to vary the file contents of different files and even of the same file used in different iterations of the same experiment. Without this, we identified another substantial source of variations – data de-duplication. If the same file contents is backed-up, even if the file name is changed then some service providers detect the duplicate data and avoid transferring it to the remote server. Interestingly, we observed this behavior both within files on the same account and between completely unrelated accounts. We discuss this effect further in Section 5.2.

We have also examined graphs of network utilization and CPU utilization across the complete backup operations and a number of interesting patterns emerge. For example, we can isolate the phase of the backup process **before** network transfer begins and observe its length and its average CPU utilization. This data suggests that some services complete CPU-intensive pre-processing work (e.g. encryption and compression where applicable) on all data up front before beginning any network transfer. This can be inefficient because it delays network transfer and can also requires substantial free disk space to store the pre-processed data (up to 1.3 times the size of the data to be backed-up in some cases.) This leads to another lesson: *Cloud storage providers should perform pre-processing tasks like encryption, compression, metadata creation, and metadata exchange incrementally and in parallel with bulk transfer of data over the network to avoid delays in network transfer and to avoid storing large amounts of temporary data.*

Several additional conclusions are our data on backup performance. *First, backing up the contents of a typical hard drive could easily take days or even weeks and during that time there will be a substantial tax on the CPU and network bandwidth of the user's machine.*[3] *Second, backup time varies substantially with factors such as size of the files, the compressibility of your data, the amount of duplicate data, etc.* Regardless, by the time a user backs up all the data on her system, she will have a substantial investment in staying with the same cloud storage provider. Once an initial back up is performed, the overhead of the backup process will be dramatically lower as only incremental backups need to be performed. As a result, end users may be loathe to switch service providers even if they are unhappy with the service provided. This makes it especially important to provide consumers with solid data on which to base their initial choice of service provider.

2.2 Restore Performance

Figure 3 reports restore performance. We report total time to restore 2.12 GB of data in the same four cases as we examined in our backup experiments – a single large file dense, a single large sparse file, many small dense files and many small sparse files.

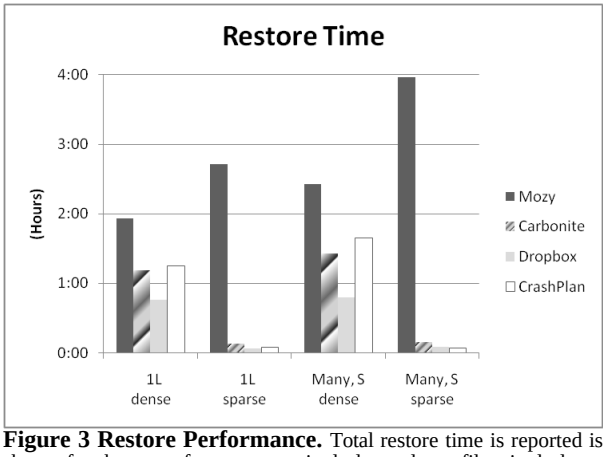

Figure 3 Restore Performance. Total restore time is reported is shown for the same four cases – single large dense file, single large sparse file, many small dense files and many small sparse files. In all cases, the total data restored is 2.12 GB. The client machine is otherwise idle during the tests.

Restore is substantially faster than backup. This is primarily due to the asymmetry of network bandwidth for upload vs. download. The bad news is that a user may still wait days or weeks to recover the contents of a typical primary hard drive. Interestingly, Dropbox regularly achieves faster download speeds and thus faster restores.

Some cloud storage providers, including Mozy and CrashPlan, are willing to send data on DVD or external harddrive via overnight mail for faster restore. However, this service is expensive (~2-4 times the cost of a yearly subscription). Users should think carefully about the amount of time they can afford to be without their data before choosing to rely on online backup.

2 We ruled out network retransmission as a substantial source of overhead by quantifying the amount of TCP retransmission from traces of network activity.

3 Some systems (e.g. Mozy and CrashPlan) offer the user some controls for minimizing the network and CPU overhead of backup at the expense of longer backup times. Others (e.g. Carbonite and Dropbox) try to pause the backup process when the machine is not idle. In the interest of space we have not presented data about the impact of these controls, but in our experience the overhead of initial backup is still substantial. For the data shown in this paper, the client machines are idle and we have chosen the settings that result in the most aggressive backups.

3. Backup Defaults and Restrictions

Cloud storage services also vary substantially in what data they backup by default. The default settings are especially important for users who want automatic backup with little to no user interaction. Table 1 lists these settings for the four providers under study [M-Defaults][C-Defaults][CP-Defaults]. Most providers designate a few directories that they will back up. Backing up important data in other directories requires some manual intervention. Mozy does an especially nice job of identifying files the user is likely to want backed-up by their file extensions, even if they are not located in a small set of default directories.

Table 1 Backup defaults

	What is backed up by default
Mozy	Identifies financial data, music, photos, documents, and other files anywhere in the file system by a list of known file extensions. Also backs up the contents of My Documents, IE Favorites, and Palm Desktop files. Videos larger than 50 MB are excluded.
Carbonite	Documents and Settings folder excluding programs, system files, temporary files, videos or individual files over 4 GB.
Dropbox	Designated Dropbox folder (typically "My Dropbox" in the My Documents folder).
CrashPlan	Documents and Settings folder.

One lesson is clear from Table 1: *Users need to manually check that all important files are indeed being backed up*. It is not safe for an end user to assume that all the data she cares about has been backed up. Unfortunately, this reduces much of the promise of automatic backup with no need for user intervention.

In most cases, users can modify the default settings manually in order to back up more files. However, in each system, there are files that cannot be backed up even by specific user request [M-Exclusions][CP-Exclusions][C-Defaults]. For example, most providers exclude system files and directories such as pagefile.sys, C:\System Volume Information and C:\Windows. Some systems are also unable to backup open files which can be a problem for important files that are left perpetually open. Dropbox will only backup files in the designated Dropbox directory. So any files users are unwilling or unable to relocate will not be backed up. Carbonite excludes temporary files.

In light of these exclusions, another lesson is clear: *Users should not view online backup services as a whole system restore solution*. For many of the circumstances that can lead to data loss, there will be substantial additional manual intervention (e.g. reinstalling the OS and applications) required to restore the system state before restoration of data from the cloud can even begin. This can be true even for software-based failures such as malware infestation.

4. Limits of Liability

In the vast majority of our tests, data was correctly restored. In fact, we never saw a case where the file contents restored were incorrect. We did, however, see one instance in which a cloud provider reported that a file back up was complete, but later was unable to restore the file at all. [4]

Regardless, reading the fine print of the providers' terms of service makes it clear that they make no guarantees about the safety of stored data [M-TOS][C-TOS][D-TOS][CP-TOS]. (This is one point on which all four service providers seem to agree.) They will do their best, but in the end, if they fail to restore data to you, then they have zero liability. We found a number of complaints of data loss in online forums.

Users could periodically restore random files to check the effectiveness of their online backup. While this "trust but verify" strategy isn't a bad idea, it would require a substantial amount of CPU and network overhead. It would also require time-consuming manual intervention or automation beyond the programming skills of most home or small business users. Such a utility could be an excellent contribution to the user community.

In many other environments, this type of risk is instead managed with insurance products. A robust market for consumer data protection insurance has not yet developed, but it is interesting to consider the role such a market could play in developing cloud computing environments. Users are in a position to place a value on their data, but not to assess likelihood of data loss based on providers management practices and track record. The actuarial and risk management arms of an insurance provider would be much better suited to track the number of data loss events per provider and to review their management practices. If a cloud storage provider did a poor job of preventing data loss, then over time the insurance rates for insuring the same data value would rise driving consumers to cloud providers with better management practices. This certainly seems like a more appropriate way to manage risks that are currently pushed to end users who are ill-equipped to assess them. However, no tools for managing risk will emerge without persistent consumer demand. We believe the systems community can play a role in educating consumers and suggesting well-structured options.

For now, the lesson we take is that *cloud storage users are wise to view cloud storage as a complement to local backups rather than as a primary backup strategy.* Restores can be slow and you may not get all the data you expect in a restore, but cloud storage should generally allow you to recover the majority of your data in the case of catastrophic loss of on-site storage. The

4 We chose not to mention the provider because the problem was not repeatable.

National Archives and Records Administration reported that 93% of businesses that lost their data center for 10 days or more due to a disaster filed for bankruptcy within one year [BackupStats]. In that light, an imperfect and inexpensive off-site backup plan may still be an excellent investment. For $60-100 per year, one might conclude “it can't hurt”, but of course, this is only true **if** the security and privacy of your data is maintained.

5. Data Privacy and Security

In this section, we explore potential risks to data privacy and security that were exposed in our testing.

5.1 Shared Key or Private Key

The good news is that all four of the cloud storage providers we tested only sent encrypted data over the network. Not doing so would clearly be disastrous for data privacy and security.

By default, the key used to encrypt the data is shared with the service provider enabling the provider to decrypt stored data at will or in response to a warrant, court order or other suitably persuasive request [Kirk10]. In addition, if data is provided in response to such requests, the service provider may not even notify the user that data has been accessed by a third party.

Shared key encryption also enables some convenient features for end users. For example, some service providers such as Mozy, Carbonite, and DropBox provide a web interface by which data backed up from one computer can be downloaded to other computers on demand. For many users, this form of remote file sharing is an especially attractive feature, allowing them for example to work on a file at home and then download it in the office.

Alternatively, providers can allow users to establish their own private key for encrypting stored data. In theory, this would prevent the service provider from accessing the contents of the user's files that are stored on their servers.

However, even if users choose their own keys, private information may be leaked. For example, encrypted files may still be associated with a file name. Filenames and information about which files are actively being modified can reveal a substantial amount of information even without access to the file contents.

Three of the four providers we tested provided a way for users to generate a private key. However, there were substantial differences in how the private key was generated, stored, and managed. Carbonite will generate a private key for you and allow you to download it. Since they generated the key in the first place, it is not particularly “private”. This feature only has value to the degree the user trusts the system not to retain a copy of the key. Mozy and CrashPlan both allow the user to generate a key themselves from a user supplied passphrase. This is a more appropriate division of responsibility. However, even if the cloud storage provider does not generate the key, they still have plenty of opportunity to record the key should they wish to do so. For example, the key is provided to the client software and could easily be transferred to the server without the user's knowledge. We are not implying that this is occurring; we are simply pointing out that in all cases, the end user must trust the cloud storage provider not to obtain/retain a copy of the key.

It is interesting (and possibly a bit paranoid or cynical) to consider what incentives a service provider might have not to retain a copy of the key. One benefit is plausible deniability in the case of a warrant or similar request for information. We suspect that most service providers would like to have such plausible deniability– either for ethical reasons or to avoid the overhead of satisfying such requests. However, there are also well-publicized examples of other companies actively participating in providing data in response to legal or governmental requests [ACLU] and it would not be difficult to modify client software to reveal the private keys when requested (i.e. the layer of plausible deniability is relatively thin). Overall, it is clear that a users' legal rights to data stored in the cloud are substantially different than their rights to data stored on their own premises. The systems community may be able to play a role in constructing technical safe-guards and advocating for legislative safe-guards.

5.2 Privacy Implications of Data De-Duplication

In our testing, we noticed that in some cases, even large dense files could be backed up almost instantly with little data transfer to the server and yet could still be restored correctly. Eventually, we isolated the source of this surprising efficiency. If a copy of the same data had been previously uploaded to the server, even under a different file name, the duplicate data would not uploaded again.

De-duplication techniques in the backed storage allow each unique piece of data to be stored only once regardless of its name or location. If you try to backup a file that the service provider has already seen (even if it was under a different name), the provider can recognize it as duplicate data, eliminating the need to actually transfer the data over the network. For the results presented in Section 2, we varied the file contents to eliminate this effect. However, in this section, we present the results of tests designed to quantify the degree of content sharing both within a single account and between accounts.

Figure 4 contains data from three tests. First, a 128 MB file with randomly generated contents is initially backed up, Second, a file with a different name, but the same contents is backed up in the same account. Finally, a third file with a different name and the same contents is backed up to a separate account. In each case, we record the amount of data sent over the network.

Service providers have a vested interest in avoiding the storage of duplicate objects and in reducing the time to back up data to their servers. Within an account, the reduction in back up time is also of value to the end user. However, cross-account is more problematic. For the service provider, this provides another source of potentially valuable data about their customers. They could easily identify related clusters of customers based on shared data. Notice that this is possible without actually viewing the contents of the files. Also, notice that cross-account sharing can occur for accounts that share the same private key. This could be used to identify multiple accounts linked to the same individual or group.

We have no reason to believe cloud service providers are currently exploiting this information, but they would certainly not be the first companies to attempt to profit by identifying related clusters of customers. Social networking sites, credit card companies, grocery stores,etc. all try to profit by identifying groups of related people for marketing purposes.

Our tests reveal that Mozy and Dropbox do content sharing both within accounts and between accounts. Carbonite does no content sharing. CrashPlan strikes a good balance; it gets the resource savings associated with intra-account sharing, but without the privacy problems of inter-account sharing.

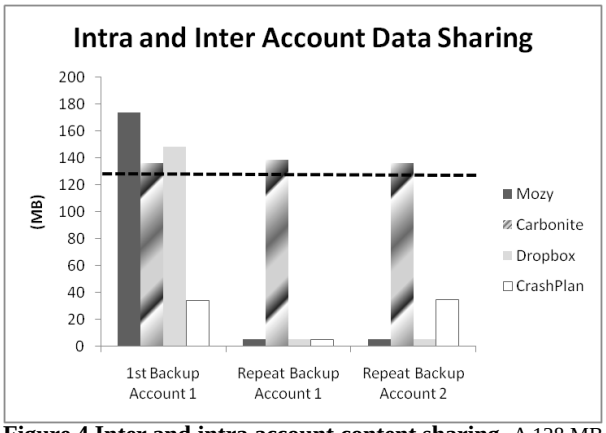

Figure 4 Inter and intra account content sharing. A 128 MB file with randomly generated contents is initially backed up, then a file with a different name but the same contents is backed up in the same account, and then a third file with a different name and the same contents is backed up to a separate account.

6. Conclusions

The data we have presented reveals some interesting differences in the way different cloud storage services are architected. Throughout the paper, we have highlighted a set of lessons to help end users evaluate whether to use cloud storage and to provide data to inform their choice of service provider. Similarly, we have pointed out where the data suggests effective architectural decisions for cloud providers.

We hope to spark a conversation about the role that the system community can play in educating cloud users, establishing standards of service for cloud storage providers, or lobbying for better risk management tools and maybe better laws. We can provide data to help users choose a cloud service provider. We could also develop tools that check if data has been backed up correctly, tools that help users manage their private keys, and tools to give increased visibility into a service provider's architecture. In general, we can help reduce the degree to which an end user must blindly trust a service provider with their data and help create the right set of incentives for cloud providers to deliver services that are well-aligned with end user needs.

7. References

[ACLU] “Phone Companies Gave Private Customer Data to Government Without Consent or Court Order, http://aclunc.org/issues/privacy.

[BackupStats] “Data Loss Statistics', Boston Computing Network, http://www.bostoncomputing.net/consultation/databackup/statistics.

[BMExtreme] BMExtreme, http://www.lp23.com/bmextreme.

[C-Defaults] “What does Carbonite back up?”, http://www.carbonite.com/how_it_works.

[C-TOS] Carbonite Terms of Service, http://www.carbonite.com/terms.

[CP-Defaults] “Changing the File Selection”, http://support.crashplan.com/doku.php/how_to/change_what_is_being_backed_up.

[CP-Exclusions] “What is Not Being Backed up”, http://support.crashplan.com/doku.php/articles/admin_excludes.

[CP-TOS] CrashPlan EULA, http://support.crashplan.com/doku.php/eula.

[Carbonite] Carbonite Online Backup, http://www.carbonite.com.

[CrashPlan], http://www.crashplan.com.

[Dropbox] Dropbox, http://www.dropbox.com.

[D-TOS] Dropbox Terms of Service, https://www.dropbox.com/terms.

[Kirk10] Marshall Kirkpatrick, “How US Government Spies Use Facebook”, ReadWriteWeb, March 16 2010.

[Mozy] Mozy Online Backup, http://www.mozy.com.

[M-Defaults] “Contents of Included Backup Sets”, Mozy Knowledge Base, Article Reference ID 3375.

[M-Exclusions] “Are there any file types or sizes that I can't back up?”, Mozy Knowledge Base, Article Reference ID 29.

[M-TOS] Mozy Terms of Service, http://mozy.com/terms.