The Eternity Service

Ross J. Anderson

Cambridge University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG
Email: ross.anderson@cl.cam.ac.uk

Abstract. The Internet was designed to provide a communications channel that is as resistant to denial of service attacks as human ingenuity can make it. In this note, we propose the construction of a storage medium with similar properties. The basic idea is to use redundancy and scattering techniques to replicate data across a large set of machines (such as the Internet), and add anonymity mechanisms to drive up the cost of selective service denial attacks. The detailed design of this service is an interesting scientific problem, and is not merely academic: the service may be vital in safeguarding individual rights against new threats posed by the spread of electronic publishing.

1 The Gutenberg Inheritance

In medieval times, knowledge was guarded for the power it gave. The Bible was controlled by the church: as well as being encoded in Latin, bibles were often kept chained up. Secular knowledge was also guarded jealously, with medieval craft guilds using oaths of secrecy to restrict competition. Even when information leaked, it usually did not spread far enough to have a significant effect. For example, Wycliffe translated the Bible into English in 1380–1, but the Lollard movement he started was suppressed along with the Peasants' Revolt.

But the development of moveable type printing by Johannes Gensfleisch zur Laden zum Gutenberg during the latter half of the fifteenth century changed the game completely. When Tyndale translated the New Testament in 1524–5, the means were now available to spread the word so quickly that the princes and bishops could not suppress it. They had him executed, but too late; by then some 50,000 copies had been printed. These books were one of the sparks that led to the Reformation.

Just as publication of the Bible challenged the abuses that had accreted over centuries of religious monopoly, so the spread of technical know-how destroyed the guilds. Reformation and a growing competitive artisan class led to the scientific and industrial revolutions, which have given us a better standard of living than even princes and bishops enjoyed in earlier centuries. Conversely, the societies that managed to control information to some extent became uncompetitive; and with the collapse of the Soviet empire, democratic liberal capitalism seems finally to have won the argument.

But what has this got to do with a cryptology conference?

Quite simply, the advance of electronic publishing has placed at risk our inheritance from Gutenberg.

Just as advancing technology in the fifteenth century made it very much harder to control information, so the advances of the late twentieth are making it very much easier. This was made clear by recent court action involving the 'Church of Scientology', one of whose former adherents had published some material which the organisation would prefer to have kept secret. This apparently included some of the organisation's 'scripture' that is only made available to members who have advanced to a certain level in the organisation.

Since Gutenberg, the publication of such a trade secret would have been irreversible and its former owners would have had to cope as best they could. However, the publication was in electronic form, so the scientologists got court orders in an action for copyright infringement and raided the primary site in the USA in August 1995. They then went to Amsterdam where they raided an Internet service provider in September, and filed for siezure of all its assets on the grounds that their copyright information had appeared on a subscriber's home page. Their next move was to raid an anonymous remailer in Finland to find out the identity of one of its users. The saga continues.

The parallel with earlier religious history is instructive. The Bible came into the public domain because once it had been printed and distributed, the sheer number of dispersed copies made it impossible for the bishops and judges and princes to gather them up for burning.

However, now that publishing has come to mean placing a copies of an electronic document on a few servers worldwide, the owners of these servers can be coerced into removing it. It is irrelevant whether the coercion comes from wealthy litigants exploiting the legal process, or from political rulers conspiring to control the flow of ideas. The net effect is the erosion of our inheritance from Gutenberg: printing is 'disinvented' and electronics document can be 'de-published'. This should concern everyone who values the benefits that have flowed from half a millenium of printing, publication and progress.

So how can we protect the Gutenberg Inheritance?

Put into the language of computer science, is there any way in which we can assure the availability of data when the threat model includes not just Murphy's ferrite beetles, the NSA and the Russian air force, but Her Majesty's judges?

2 Preventing Service Denial

This problem is merely an extreme case of a more general one, namely how we can assure the availability of computerised services. This problem is one of the traditional goals of computer security, the others being to assure the confidentiality and integrity of the information being processed.

Yet there is a strange mismatch between research and reality. The great majority of respectable computer security papers are on confidentiality, and almost all the rest on integrity; there are almost none of any weight on availability.

But availability is the most important of the three computer security goals. Outside the military, intelligence and diplomatic communities, almost nothing is spent on confidentiality; and the typical information systems department in civil government or industry might spend 2% of its budget on integrity, in the form of audit trails and internal auditors. However 20-40% of the budget will be spent on availability, in the form of offsite data backup and spare processing capacity.

There are many kinds of record that we may need to protect from accidental or deliberate destruction. Preventing the powerful from rewriting history or simply suppressing embarrassing facts is just one of our goals. Illegal immigrants might wish to destroy government records of births and deaths[1]; real estate owners might attack pollution registries; clinicians may try to cover up malpractice by shredding medical casenotes [Ald95]; fraudsters may 'accidentally' destroy accounting information; and at a more mundane level, many computer security systems become vulnerable if audit trails or certificate revocation lists can be destroyed.

There is also the problem of how to ensure the longevity of digital documents. Computer media rapidly become obsolete, and the survival of many important public records has come under threat when the media on which they were recorded could no longer be read, or the software needed to interpret them could no longer be run [Rot95].

For all these reasons, we believe that there is a need for a file store with a very high degree of persistence in the face of all kinds of errors, accidents and denial of service attacks.

3 Previous Work

Many papers purport to show that the average firm could not survive long for without its computers, and that only 20–40% of firms have properly tested disaster recovery plans. The authors of such papers conclude that the average firm will not survive when a disaster strikes, and that company directors are thus being negligent for not spending more money on disaster recovery services. The more honest of these papers are presented as marketing brochures for disaster recovery services [IBM93], but many have the appearance of academic papers.

They are given the lie by incidents such as the Bishopsgate bomb in London where hundreds of firms had systems destroyed. Some banks lost access to their data for days, as both their production and backup sites were within the 800 yard police exclusion zone [Won94]. Yet we have no report of any firm's going out of business as a result. A more recent IRA bomb in London's dockland area confirmed the pattern: it also destroyed a number of computer installations, yet companies bought new hardware and recovered their operations within a few days [Bur96].

[1] The population of California is said to have increased significantly after fire destroyed San Francisco's birth records in the wake of the great earthquake.

So we can ignore most of the existing literature on availability, and indeed we have to look rather hard for respectable papers on the subject. One of the few of which we are aware [Nee94] suggests that availability has to do with anonymity — anonymous signalling prevents denial of service attacks being selective. That insight came from studying burglar alarm systems, and it also makes sense in our publication scenario; if the physical location of the worldwide web site cannot be located, then the rich man's lawyers will have nowhere to execute their seizure order. But how could an anonymous publication service be realised in practice?

4 The Eternity Service

We draw our main inspiration from the Internet, which was originally conceived to provide a communications capability that would survive a global thermonuclear war. Is it possible to build a file store which would be similarly resilient against even the most extreme threat scenarios?

Firstly, let us sketch a high level functional specification for such a store, which we will call the 'Eternity Service'[2].

4.1 What it does

The Eternity Service will be simple to use. Say you want to store a 1MB file for 50 years; there will be a tariff of (say) $99.95. You upload a digital coin for this, together with the file; no proof of identity or other formality is needed. After a while you get an ack, and for the next 50 years your file will be there for anyone to get by anonymous file transfer.

Copies of the file will be stored on a number of servers round the world. Like the Internet, this service will depend on the cooperation of a large number of systems whose only common element will be a protocol; there will be no head office which could be coerced or corrupted, and the diversity of ownership and implementation will provide resilience against both error and attack.

The net effect will be that your file, once posted on the eternity service, cannot be deleted. As you cannot delete it yourself, you cannot be forced to delete it, either by abuse of process or by a gun at your wife's head.

External attacks will be made expensive by arranging things so that a file will survive the physical destruction of most of the participating file servers, as well as a malicious conspiracy by the system administrators of quite a few of them. If the servers are dispersed in many jurisdictions, with the service perhaps even becoming an integral part of the Internet, then a successful attack could be very expensive indeed — hopefully beyond even the resources of governments.

[2] In 'The City and the Stars', Arthur C Clarke relates that the machinery of the city of Diaspar was protected from wear and tear by 'eternity circuits'; but he omits the engineering details.

The detailed design will utilise the well known principles of fragmentation, redundancy and scattering. But before we start to consider the details, let us first consider the threat model.

4.2 The threat model

Perhaps the most high level threat is that governments might ban the service outright. Might this be done by all governments, or at least by enough to marginalise the service?

The political arguments are quite predictable. Governments will object that child pornographers, Anabaptists and Persian spies will use the service, while libertarians will point out that the enemies of the state also use telephones, faxes, email, video and every other medium ever invented. Software publishers will be afraid that a pirate will Eternally publish their latest release, and ask for an 'escrow' facility that lets a judge have offending matter destroyed; libertarians will object that no judge today can destroy the information contained in a personal advertisement published in 'The Times' at the cost of a few pounds.

But law tends to lag technology by a decade or more; it is be hard to get all governments to agree on anything; and some countries, such as the USA, have free speech enshrined in their constitutions. So an effective worldwide ban is unlikely. There might always be local bans: Israeli agents might put up a file containing derogatory statements about the Prophet Mohammed, and thus get eternity servers banned in much of the Muslim world. If it led to a rejection of the Internet, this might provide an effective attack on Muslim countries' ability to develop; but it would not be an effective attack on the Eternity Service itself, any more than the Australian government's ban on sex newsgroups has any effect on the US campuses where many of the more outré postings originate.

Most non-legislative global attacks can be blocked by technical means. Network flooding can never be completely ruled out, but can be made very expensive and unreliable by providing many access points, ensuring that the location of individual files remains a secret and integrating the service with the Internet.

So in what follows, we will focus on the mechanisms necessary to prevent selective service denials at finer levels of granularity. We will imagine that an ignorant or corrupt judge has issued an injunction that a given file be deleted, and we wish the design of our system to frustrate the plaintiff's solicitors in their efforts to seize it. We will also imagine that a military intelligence agency or criminal organistion is prepared to use bribery, intimidation, kidnapping and murder in order to remove a file; our system should resist them too. The basic idea will be to explore the tradeoffs between redundancy and anonymity.

4.3 A simple design

The simplest design for an eternity service is to mimic the printed book. One might pay 100 servers worldwide to retain a copy of the file, remember the names of a randomly selected 10 of them (to audit their performance and thus enforce the contract), and destroy the record of the other 90.

Then even if the user is compelled by authority to erase the file and to hand over the list of ten servers where copies are held, and these servers are also compelled to destroy it, there will still be ninety surviving copies scattered at unknown locations round the world. As soon as the user escapes from the jurisdiction of the court and wishes to recover his file, he sends out a broadcast message requesting copies. The servers on receiving this send him a copy via a chain of anonymous remailers.

Even if the protection mechanisms are simple, the use of a large number of servers in a great many jurisdictions will give a high degree of resilience.

4.4 The perjury trap

Significant improvements might be obtained by intelligent optimisation of the legal environment. For example, server should not delete eternity files without manual approval from a security officer, whose logon procedure should require him to declare under oath that he is a free agent, while the logon banner states that access is only authorised under conditions of free will.

Thus, in order to log on under duress, he would have to commit perjury and (in the UK at least) contravene the Computer Misuse Act as well. Courts in most countries will not compel people to commit perjury or other criminal offences.

We refer to this protection measure as a ‘perjury trap’. It might be useful in other applications as well, ranging from root logon to general systems to the passphrases used to unlock decryption and signature keys in electronic mail encryption software like PGP.

4.5 Using tamper-proof hardware

Using a perjury trap may block coercion of the abuse-of-process kind in many countries, but we must still consider more traditional kinds of coercion such as kidnapping, extortion and bribery.

In order to protect the owner of the file from such direct coercion, we have the rule that not even the owner may delete a file once posted. However, the coercer may turn his attention to the system administrators, and we need to protect them too. This can best be done if we arrange things so that no identifiable group of people — including system administrators — can delete any identifiable file in the system.

The simplest approach is to encapsulate the trusted computing base in tamper-resistant hardware, such as the security modules used by banks to protect the personal identification numbers used by their customers in autoteller machines [JDK+91]. Of course, such systems are not infallible; many of them have failed as a result of design errors and operational blunders [And94], and even if keys are kept in specially hardened silicon chips there are still many ways for a wealthy opponent to attack them [BFL+93].

However, given wide dispersal as one of our protection mechanisms, it may be too expensive for an opponent to obtain and break a quorum of tamper resistant devices within a short time window, and so the combination of tamper resistance with careful protocol design may be sufficient. In that case, the Eternity Service could be constructed as follows.

Each hardware security server will control a number of file servers. When a file is first loaded on to the system, it will be passed to the local security server which will share it with a number of security servers in other jurisdictions. These will each send an encrypted copy to a file server in yet another jurisdiction.

When a client requests a file that is not in the local cache, the request will go to the local security server which will contact remote ones chosen at random until one with a copy under its control is located. This copy will then be decrypted, encrypted under the requester's public key and shipped to him.

Communications will be anonymised to prevent an attacker using traffic analysis to link encrypted and plaintext files. Suitable mechanisms include mix-nets (networks of anonymous remailers) [Cha81] and rings [Cha88]. The former are suitable for sending the file to the user, and the latter for communications between security servers; even traffic analysis should not yield useful information about which file server contains a copy of which file, and this may be facilitated by traffic padding [VN94].

Note that the existence of secure hardware allows us to substantially reduce the number of copies of each file that have to be kept. It is sufficient that the attacker can no longer locate all copies of the file he wishes to destroy. Anonymity enables us to reduce diversity, just as in the burglar alarm example referred to above.

4.6 Mathematics or metal?

Relying on hardware tamper resistance may be undesirable. Firstly, it is relative, and erodes over time; secondly, export controls would slow down the spread of the system; and, thirdly, special purpose low-volume hardware can be expensive. Now it is often the case that security properties can be provided using mathematics rather than metal. Can we use mathematics to build the eternity service?

Protecting the location of file copies means that location information must be inaccessible to every individual user, and indeed to every coercible subset of users. Our goal here is to use techniques such as threshold decryption and Byzantine fault tolerance, as implemented in Rampart [Rei94].

Byzantine fault tolerance means, for example, that with seven copies of the data we can resist a conspiracy of any two bad sysadmins, or the accidental destruction of four systems, and still make a complete recovery. Using Byzantine mechanisms alone, incomplete recovery would be possible after the destruction of up to six systems, but then there would be no guarantee of integrity (as such a ‘recovery’ could be made by a bad sysadmin from bogus data).

There are some interesting interactions with cryptography. If all files are signed using a system key, then a full recovery can still be made so long as there is just one surviving true copy of the file in the system, and the public key is not subverted. Of course, it is rare to get something for nothing, and we must then make it hard to compromise the signing key (and feasible to recover from such a compromise).

We will need to provide for in-service upgrades of the cryptographic mechanisms: progress in both cryptanalysis and computer engineering may force the adoption of new signature schemes, or of longer keylengths for existing ones. We will also need to recover from the compromise of any key in the system.

Users may also want to use cryptography to add privacy properties to their files. In order to prevent a number of attacks (such as selective service denial at retrieve time) and complications (such as resilient management of authentication), the eternity service will not identify users. Thus it cannot provide confidentiality; it will be up to users to encrypt data if they wish and are able. Of course, many users will select encryption schemes which are weak, or which become vulnerable over time; and it may be hoped that this will make governments less ill-disposed towards the service.

4.7 Indexing

The system’s directory will also have to be a file in it. If users are left to remember file names, then the opponent can deny service by taking out an injunction preventing the people who know the name from revealing it.

The directory should probably contain not just the file’s logical name (the one which relevant security servers would understand), but also some further labels such as a plaintext name or a keyword list, in order to allow retrieval by people who have not been able to retain machine readable information.

The current directory might be cached locally, along with the most popular files; in the beginning, at least, the eternity service may be delivered by local gateway servers. Injunctions may occasionally be purchased against these servers, just as some university sites censor newsgroups in the `alt.sex.*` namespace; however, users should still be able to ftp their data from overseas gateways. Ultimately, we will aim for a seamless integration with the rest of the Internet.

4.8 Payment

The eternity service may have to be commercialised more quickly than the rest of the Internet, as storage costs money paid locally, while most academic network costs are paid centrally. Here we can adapt digital cash to generate an ‘electronic annuity’ which follows the data around.

Provided the mechanics can be got right, the economics will get better all the time for the fileserver owners — the cost of disk space keeps dropping geometrically, but they keep on getting their $1 per MB per year (or whatever) for their old files. This will motivate server owners to guard their files well, and to copy them to new media when current technology becomes obsolete.

But the confidentiality properties needed for electronic annuities are not at all straightforward. For example, we may want banks to underwrite them, but we do not want the opponent's lawyers enjoining the bankers. Thus the annuity will probably need to be doubly anonymous, both for the client vis-à-vis the bank and for the bank vis-à-vis the network. How do we square this with audit and accountability, and with preventing money laundering? What if our bent judge orders all banks to delay payment by long enough for the financier of an allegedly libellous file to be flushed out? These requirements do not seem to have been tackled yet by digital cash researchers.

Another problem will arise once the service becomes profitable. Presumably there will be a market in revenue-generating Eternity servers, so that a fileserver owner who wishes to cash in and retire can sell his revenue generating files to the highest bidder. The obvious risk is that a wealthy opponent might buy up enough servers to have a significant chance of obtaining all the copies of a target file. The secondary risk is that a single network service provider might acquire enough market share to penetrate the anonymity of communications and track down the copies.

How can these risks be controlled? One might try to certify server owners, but any central body responsible for certifying 'this site is not an NSA site' could be bought or coerced, while if the certification were distributed among many individuals, few of them would have the resources to investigate would-be server owners thoroughly. An alternative could be to leave the security policy to the user who uploads the file: she could say something like, 'I want seven copies of my file to be moved randomly around the following fifty sites'. The problem here is how we prevent policy erosion as sites are replaced over time.

At a more mundane level, we need mechanisms to stop a file server owner cheating by claiming annuity payments on a file without keeping a copy all the time. After all, he could just download the file from the Eternity Service itself whenever he needs to demonstrate possession. This provides yet another reason why files must be encrypted with keys the server owners do not know; then the annuity payment server can pose a challenge such as 'calculate a MAC on your file using the following key' to check that the annuitant really has kept all the data that he is being paid to keep.

4.9 Time

One of the complications is that we need to be able to trust the time; otherwise the opponent might manipulate the network time protocol to say that the date is now 2500AD and bring about general file deletion. Does this bring the

Network Time Protocol (and thus the Global Positioning System and thus the US Department of Defense) within the security perimeter, or do we create our own secure time service? The mechanics of such a service have been discussed in other contexts, but there is as yet no really secure clock on the Internet.

A dependable time service could benefit other applications, such as currency exchange transactions that are conducted in a merchant's premises while the bank is offline. Meanwhile, we must plan to rely on wide dispersal, plus some extra rules such as 'assets may not be deleted unless the sysadmin confirms the date', 'the date for deletion purposes may never exceed the creation date of the system software by five years', and 'no file may be deleted until all annuity payments for it have been received'.

5 Conclusion

The eternity service that we have proposed in outline here may be important in guaranteeing individual liberties against the abuses of power. It is also interesting from the scientific point of view, and the purpose of this paper has been to present it to the cryptology and computer security communities as an interesting problem that merits further study.

Building the eternity service will force us to clarify a number of points such as the nature of secure time, the limits to resilience of distributed authentication services, and the write-once indexing of large databases. The project should also broaden our understanding of anonymity. It appears, for example, that the difficulty of scaling anonymous communications is an essential feature rather than a nuisance; if there were just one channel, the judge could have it cut or flooded.

Perhaps the most interesting aspect of the service is that it might teach us a lot about availability. Just as our appreciation of confidentiality was developed by working out the second- and third-order effects of the Bell LaPadula policy model [Amo94], and authenticity came to be understood as a result of analysing the defects in cryptographic protocols [AN95], so the Eternity Service provides a setting in which availability services must be provided despite the most extreme opponents imaginable.

Acknowledgements

Some of these ideas have been sharpened in discussions with Roger Needham, David Wheeler, Matt Blaze, Mike Reiter, Bruce Schneier, Birgit Pfitzmann, Peter Ryan and Rajashekhar Kailar; and I am grateful to the Isaac Newton Institute for hospitality while this paper was being written.

References

[Ald95] "Nurse sacked for altering records after baby's death", K Alderson, *The Times* 29 November 95 p 6

[Amo94] *'Fundamentals of Computer Security Technology'*, E Amoroso, Prentice Hall 1994

[And94] "Why Cryptosystems Fail" in *Communications of the ACM* vol 37 no 11 (November 1994) pp 32–40

[AN95] RJ Anderson, RM Needham, "Programming Satan's Computer", in *'Computer Science Today — Recent Trends and Developments'*, J van Leeuven (ed.), Springer Lecture Notes in Computer Science volume 1000 pp 426–440

[Bur96] "Rising from the Rubble", G Burton, in *Computer Weekly* (29 Feb 1996) p 20

[BFL+93] S Blythe, B Fraboni, S Lall, H Ahmed, U de Riu, "Layout Reconstruction of Complex Silicon Chips", in *IEEE J. of Solid-State Circuits* v 28 no 2 (Feb 93) pp 138–145

[Cha81] D Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", in *Communications of the ACM* v 24 no 2 (Feb 1981) pp 84–88

[Cha88] D Chaum, "The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability", in *Journal of Cryptology* v 1 (1988) pp 65–75

[IBM93] *'Up the creek? — The business perils of computer failure'*, IBM, 1993

[JDK+91] DB Johnson, GM Dolan, MJ Kelly, AV Le, SM Matyas, "Common Cryptographic Architecture Application Programming Interface", in *IBM Systems Journal* **30** no 2 (1991) pp 130 - 150

[Nee94] RM Needham, "Denial of Service: an Example", in *Communications of the ACM* v 37 no 11 (Nov 94) pp 42–46

[Rei94] MK Reiter, "Secure Agreement Protocols: Reliable and Atomic Group Multicast in Rampart", in *Proc. ACM Conf. on Computer and Communications Security 1994* pp 68–80

[Rot95] J Rothenberg, "Ensuring the Longevity of Digital Documents", in *Scientific American* (January 1995) pp 24–29

[VN94] BR Venkataraman, RE Newman-Wolfe, "Performance Analysis of a Method for High Level Prevention of Traffic Analysis Using Measurements from a Campus Network", in *Computer Security Applications 94* pp 288–297

[Won94] K Wong, "Business Continuity Planning", in *Computer Fraud and Security Bulletin* (April 94) pp 10 - 16

This article was processed using the LaTeX macro package with LLNCS style