Protecting Free Speech using the Eternity Service

Adam Johnson

June 2003

Abstract

The Eternity Service is an idea of Ross J. Anderson, and is a service where people can post potentially controversial or censored documents or other data anonymously, thus providing people with a powerful vehicle for true freedom of speech.

The Service guarantees not just anonymity but also that the document will be available for as long as was asked for. Thus giving electronic publishing the same, if not more, strength of permanence as the printing press.

1 Introduction

Most people know that the Internet was built to survive Thermonuclear War and therefore that it is very robust. But this is only infrastructure, the communications network, there is almost nothing to protect the content. It is possible for a single person to remove everyone's access to a publication through physical, legal or technological attack.

A recent example off this sort of attack is where al-Jazeera the Arabic news broadcaster's English website was made unavailable for weeks by a

single person tricking the DNS administrators[Bro03], most likely because they disagreed with the American prisoners of war being shown on TV.

With the main Governments of the western world, especially America, reducing the freedoms of their public due to knee-jerk reactions to the rise in terrorism it will be desirable and maybe even necessary to have a way to ensure that people can publish their opinions without undeserved consequences from others.

The Eternity service first described by Ross J. Anderson is a way to ensure that anyone, including the public, can voice their opinion anonymously and safely without being censored. I will attempt to bring some of the ideas discussed in the more well known articles to the attention of the reader, including Tonda Benés's paper covering in more detail the encryption schemes[Ben01] required for the service and Ian Clark's beta implementation Freenet[CH+02].

2 Free speech

> *Everyone values their freedom, in fact, many consider it so important that they will die for it. People like to think that they are free to form and hold whatever opinions they like, particularly in western countries. Consider now that someone had the ability to control the information you have access to. This would give them the ability to manipulate your opinions by hiding some facts from you, by presenting you with lies and censoring anything that contradicted those lies. This is not some Orwellian fiction, it is standard practice for most western governments to lie to their populations, so much so, that people now take it for*

> *granted, despite the fact that this undermines the very democratic principles which justify the government's existence in the first place.* - Ian Clarke[CH+03b]

Free Speech has been recognized as incredibly important by philosophers, political scientists and law makers for hundreds of years. Now that a large part of the world is largely run by corporations and one persons point of view can be heard around the globe, people in higher positions are trying to reduce what people can say about them, our right to free speech is beginning to be lessened.

In Charles Pfleeger's book *Security in Computing*[Pfl97] he suggests a small set of basic moral principles. 'The right to know' that is the right to know about your rights, the truth about people and corporations you do business with and to know the truth of what is happening in your world.

2.1 The importance of anonymity

The second principle that Pfleeger mentions is 'The right to privacy', an individual has the right to keep information about themselves and their opinions private, unless it interferes with others right to know.

One of the easiest and most effective forms of censorship has always been to punish people after they have said or published something that those with the power of censoring didn't agree with. The only way to protect people from this extreme, and unfortunately violent in some oppressive countries, from of censorship is to protect their identity.

Keeping people's identities private enables and empowers people to speak about what they believe in, since there is no fear of their rights being taken away.

3 The Eternity Service

The Eternity service is a large, preferably global, network of servers that store digital documents and other data securely for a set period of time. There is no direct way to delete data and all communications are performed anonymously using mix-nets (groups of anonymous remailers). Users could access the service with a client program designed to request documents and insert data. When data was inserted into the network it would be randomly distributed to several servers in the network.

So that the data can be located and compared a hash of the data is created. This is used for validating the data and for searching for it. However since a hash value isn't a very user friendly way of locating and describing data, there will be a human readable namespace in which users can link a documents hash value to a global namespace similar to the current Internets URLs.

In the proposed design by Ross J. Anderson[And96] a digital cash implementation is required because the user placing data in the service pays for the storage space in advance. The monetary transfers are done anonymously and the server has to prove that it has stored the document as the bank pays out per year.

In Ian Clarke's et al.[CH+03a] implementation the users don't have to pay for storage space. The project requires the goodwill of the users to provide a portion of their disk space to the network. Freenet doesn't have specified lengths of time for the documents to be stored, their lifetime is based on how popular the document is.

3.1 Threats

Such a large ambitious network will have to deal with many different threats. Possibilities include:

- Hardware Failure

  The only way to really protect a system from hardware failure is to have built in redundancy. Running servers with redundant storage will probably be a good idea if they are popular nodes, or you could just rely on the replication of data across the network.

- Oppressive Governments

  There will always be governments that oppose this kind of uncontrollable free speech. There is no practical way to ensure availability of the service if the government choose to block access to all but specific parts of the Internet (ie. filtered websites as is the case in mainland China) or the entire Internet (as in North Korea). But one would hope that the people would stand up to such a government.

- Legal threats

  If a document is published that say a particular corporation or religious cult didn't approve of then they would most likely try to use the law to shut down the server that provides the document. This is why it is important for the publisher to be anonymous and also to not know which server is providing the document

- Script Kiddies and Crackers

  There will always be people that will try to break a system just for the fun of it or for bragging rights. The only real defense will be to make the encryption mathematically strong enough, and to make systems so

difficult to compromise as to avoid traffic analysis, or a large number of corrupt servers.

- Spammers

  With no way of removing documents from the service overloading it with useless data may be an effective attack. Freenet could deal with this type of attack since spam will never be that popular so it will expire quickly. Neither Anderson or Benés have addressed this sort of threat or how they would deal with it in their analysis.

4 How to build it

How would such a system be built. How do you ensure the anonymity of the participants in the network. This is no easy task and is the topic of much current research and academic discussion.

4.1 Architecture

> *Without anonymity there can never be true freedom of speech, and without decentralization the network will be vulnerable to attack.* - Freenet Project[CH+03a]

When Anderson published his paper on the Eternity Service in 1996 the peer-to-peer style design wasn't very common or well discussed, so the original service is designed with the more traditional client-server pattern instead, with no discussion of the merits of each design.

With a client-server design it is easier to be efficient since there are no issues with routing information and bandwidth use is much less than with peer-to-peer style design. However, even with a large number of dedicated

Eternity Services, the design is still reasonably centralised and is easier to find the main points of weakness and disable the service.

With a peer-to-peer design the size of the network can be exceedingly large. But this can be used to advantage: if every user donates a small part of their disk space then then the total storage space can become astonishingly large; the amount of traffic increases greatly which makes it easier to keep communications anonymous and makes traffic analysis much more difficult.

To summarize the differences:

- Client Server

  A Client server design would have a large amount of dedicated servers running around the globe. Each server would be aware of the existence of a significant number of other servers. They would communicate and send requests using mix-nets (discussed below).

  Users would use a client program that located a server and would communicate to the rest of the service through that connection. The commercial aspect of the service would have to be used to make the service feasible due to the cost of running these dedicated servers.

- Peer to Peer

  A Peer to Peer design wouldn't require a known group of dedicated servers. Instead any person that wanted to use the service would run a program known as a node that would communicate with other nodes around it, accepting requests and sharing some of that machines disk space with the rest of the service.

  The original design of mix-nets (discussed below) couldn't be used for communication because of the need for dynamic routing (since it is possible that a node could leave the network at any time) so the peer

to peer design isn't as well protected from traffic analysis style attacks as well as the dedicated server model

4.2 Public key cryptography

Public key cryptography is a very important part of the service. Each server or node of the service requires a public-private key pair so that it can communicate with the rest of the network securely. The private key is kept secret and is never shown to any other node of the network or otherwise that nodes security is lost. The public key is available to any node that needs or cares to know it doesn't need to be secret.

The effectiveness of public key cryptography is in the mathematical symmetry of the algorithm. A message encrypted with a servers public key can only be decrypted with the corresponding private key, this way a message can be sent to that node and it is the only one capable of reading it. If on the other hand a message is encrypted with that nodes private key, then it can be decrypted with the public key.

This enables any two nodes to communicate securely by using the public key of the destination node to encrypt the message, and vice versa for the other node.

This also allows the user to digitally sign the data with a certain key. For example a user could create a public-private key pair and let the public key be known. Then any document this person cares to publish can be encrypted with their private key. Anyone can decrypt the document using the public key of this particular user so a reputation and trust can be built up with out users ever knowing who each other is.

4.3 Mix nets

Mix-nets are a simple idea but require a few tricks to make them provably anonymous. I will explain sender anonymity here, but it is also possible to make the receiver anonymous[Cha81].

The sender knows the address of the server to receive the message and chooses a route of randomly chosen mix servers to the recipient. The sender then encrypts the message using the public key of the last server on the route (and possibly pads the message with some random data to prevent identification of the message). That message plus the address of the destination is then encrypted with the public key of the server just prior in the randomly chosen route. For each mix server in the route the address of the next server in the route is added and then encrypted, using the public key of that server. This results in an encrypted chain of address similar to the way messages are encapsulated in ordinary network protocols.

The sender then sends the entire message to the first 'Mix server' on the route. This server only knows who sent the message and once it decrypts the message it now knows the address to send it to. It doesn't know if the previous sender was the originator or if the next Mix server in the route is the final destination. It then sends the resulting message to the address of the server in the message it received.

When the message reaches its destination it does so without any of the other servers along the route knowing what the final destination was. Using this method the Eternity servers of nodes can send requests and transmit data without revealing their identity to the rest of the network

5 Conclusion

With the ability to communicate securely using public key encryption and anonymously using Mix-nets, it is quite feasible to create a network that replicates data. One that is almost invulnerable to most forms of attack and that provides us with a vehicle for one of the most important rights that we may quite possibly loose within the coming years. We should take advantage of the ability to study this exciting field while we are still able to.

6 Acknowledgments

I would like to thank Shan Lun and Paul Shotbolt for their important comments and critiques while writing this paper, and especially Ana Stilinovic for her corrections and help in making my writing understandable.

References

[And96] Ross J. Anderson. *The Eternity Service*
Proceedings of Pragocrypt 1996.

[Ben01] Tonda Benés. *The Strong Eternity Service*
Information Hiding, 4th International Workshop 2001.

[Bro03] Russell Brown. *Dissenters silenced*
NZ Listener, Volume 188 Number 3287 May 2003.

[Cha88] David Chaum. *The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability*
Journal of Cryptology, Volume 1 Number 1 1988.

[Cha81] David Chaum. *Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms*
Communications of the ACM, Volume 24 Number 2 1981.

[CH+03a] Freenet Project. *The Free Network Project*
http://freenet.sourceforge.net

[CH+03b] Ian Clarke, et al. *The Philosophy behind Freenet*
http://freenet.sourceforge.net/index.php?page=philosophy

[CH+02] Ian Clarke, Theodore W. Hong, Scott G. Miller, Oskar Sandberg, Brandon Wiley. *Protecting Free Expression Online with Freenet*
IEEE Internet Computing, Volume 6 Number 1 2002.

[Pfl97] Charles P. Pfleeger. *Security in Computing* - Second Edition
Prentice Hall PTR, 1997.