Position Summary: Censorship Resistant Publishing Through Document Entanglements

Marc Waldman and David Mazières
Computer Science Department, NYU
{waldman,dm}@cs.nyu.edu

Today, most documents available over the the Internet are easy to censor. Each document can usually be traced back to a specific host or even the individual responsible for publishing the document. Someone wishing to censor a document can use the courts, threats, or some other means to force the host administrator or author to delete a particular file. Certainly, there are some high profile documents that are widely mirrored across the internet, however this is not an option for most published documents.

Clearly, a censorship resistant system must replicate a published document across many hosts. However, no standard naming convention exists that allows one to easily specify several hosts via a single name. Even if such a naming convention existed it merely makes the censor's job somewhat harder — the censor still knows exactly which hosts contain the content and therefore which hosts to attack.

Currently, there is little incentive or justification for a server administrator to store documents that he is being pressured into deleting. We propose a system, named Tangler, that we believe provides some incentive to retain such documents and solves the document naming problem.

Tangler is a censorship resistant distributed file system that employs a unique document storage mechanism. A group of documents, called a collection, can be published together under a single name. This collection is named by a public key, K. Only the individual possessing K's corresponding private key can publish a collection under the name K. By naming the published collections in a host and content independent manner we allow the publisher to securly update the collection at some point in the future. This naming convention also allows a collection to include pointers, called soft links, to other collections. These collections may be owned and updated by other individuals.

In order to publish a collection, C, one runs a program that fetches random blocks of previously published collections and $entangles$ these blocks with those of C. Once $entangled$, collection C is dependent on the randomly chosen blocks. Without these blocks, collection C cannot be reassembled. Therefore the publisher has some incentive to retain these blocks, some of which belong to other collections. Notice that this implies that each block may belong to several collections and that each block can be used to reassemble more than one collection.

Tangler's local caching policy causes the replication of these dependent blocks which, at some point in the future, may be reinjected into the distributed file system. This caching policy and reinjection mechanism helps make the published collection difficult to censor.

Tangler consists of a dynamic group of file servers that can publish documents to a distributed file system. Each file server donates local disk space to the system. Servers can join or leave the system at will. The participating servers collectively form a MIX based network that is used for untraceable communication among the servers.

Our block $entanglement$ algorithm is based on Shamir's secret sharing scheme. In this scheme a secret, s, can be split into n pieces, called shares, such that any $k \leq n$ of them can be combined to form s. In our case s is the collection block we wish to $entangle$.

The $entangle$ algorithm takes three parameters, a collection block b and two blocks from previously publish collections. Call these two blocks p_1 and p_2 respectively. These two blocks will become $entangled$ with b. Block b will therefore depend on p_1 and p_2. Each block has the same format; it essentially consists of an x and y value. The points implied by p_1, p_2 and $(0, b)$ uniquely define a quadratic equation. This quadratic is then evaluated at two random x values. This produces two new blocks which we will call q_1 and q_2. Blocks p_1, p_2, q_1 and q_2 are cached and copied to the distributed file system. Notice that we have not cached or copied block b. Block b can be reconstructed from any three of the four stored blocks, namely p_1, p_2, q_1 or q_2.

The $entanglement$ process has defined a (3,4) threshold secret sharing scheme where 3 of any 4 shares can recover the secret (block b). Our publish algorithm is essentially Shamir's secret sharing scheme with a slight twist. Rather than randomly selecting the coefficiants of a quadratic equation we create the quadratic equation from blocks of published collections and the block we wish to publish.