Tamias: a privacy aware distributed storage

Jean Lorchat, Cristel Pelsser, Randy Bush, Keiichi Shima
Internet Initiative Japan, Inc.
`{jean,cristel,randy,keiichi}@iijlab.net`

1 Introduction

Today's distributed storage solutions do not enable fine sharing and privacy control to their users. Here we describe Tamias: a distributed storage system that can accommodate both needs to **share** data and keep them **private**, while providing users with detailed controls over the sharing process. To achieve this, we introduce a fine-grained access control infrastructure on top of an existing storage system.

Contrary to the *Web 2.0* trends and more in line with the *Freedom in the Cloud*[1] movement, we want to ensure that we do not give storage access to untrusted third parties, though we wish to share and distribute storage. So, should we need to use such third parties, we want to not trust them with our precious data, but rather give them an undecipherable version at best.

With privacy and security as our main focus, we believe that a distributed solution can offer better scalability, resilience, and independence from centralized solutions which would require trust in a centralized storage provider.

2 Solution overview

We chose Tahoe-LAFS[2] (a free and open storage system) as the foundation for our storage system because it provides a few very interesting properties: (i) Full encryption on the user device allowing lack of trust in both the storage provider and the transport network; separate read, write and content verification access. (ii) Erasure coding for increased resiliency and performance. (iii) Self-granting access capabilities for decentralized access control.

Building on these strong points, we believe we will be able to create the privacy-aware storage that we desire, assuming that we add the following functionalities to Tahoe: (1) User identification and authentication. (2) Capability signing and encryption. (3) A user-centric repository for in-band exchange of URIs.

Our solution to provide these features uses a buddy list that acts as a repository of public keys of "friends". Public-key cryptography coupled with user authentication and authorization allows us to extend capabilities with information about intended recipient identity and owner signing, an important step to achieving fine grained (per-object/per-user) access control. The public keys also serve as unique identifiers for each user.

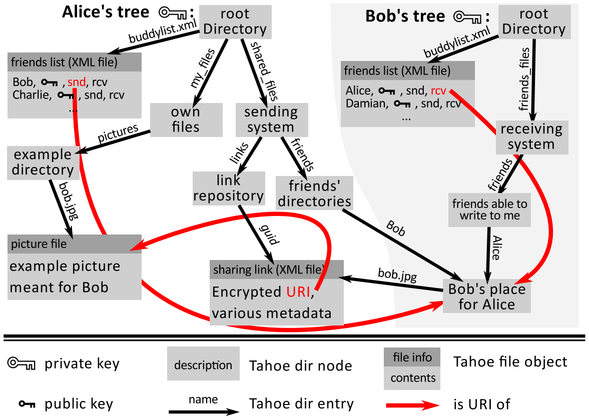

Figure 1: Private sub-tree stored within Tahoe

The buddy list is stored as a Tahoe object in a dedicated sub-tree, making our modification self-contained. The access capability to this tree is derived from the user's private key, making it possible to get a consistent view across multiple devices by simply configuring the private key on each device.

The tree holds all the information required to control the sharing process, as described in figure 1. An obvious advantage of this solution is that it requires changes on the client only, requiring less modification of Tahoe for the moment.

3 Summary and future work

We are adding public-key cryptography extensions to the Tahoe storage system so that file owners can be identified, and so that these owners may control with whom and how they choose to share files.

With the primary goal of privacy achieved, we will need to addresses the issues of delegation and revocation. The former is easy to achieve by re-signing certified capabilities. The latter will involve classic time-based solutions and the use of revocation services.

Finally, as any software using keys, it will need to integrate well with existing key management software in order to be successful, an important one being pgp/gpg.

References

[1] Eben Moglen, http://www.softwarefreedom.org/events/2010/isoc-ny/FreedomInTheCloud-transcript.html, Freedom in the Cloud (transcript), Feb. 2010

[2] http://tahoe-lafs.org : Tahoe-LAFS homepage