TAMIAS: a privacy aware distributed storage

Jean Lorchat, Cristel Pelsser, Randy Bush, Keiichi Shima
Internet Initiative Japan - Research Laboratory

The situation today

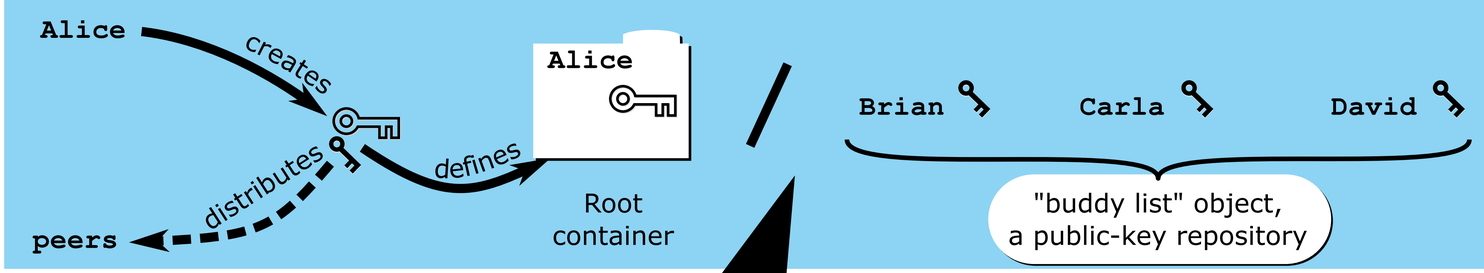
Many providers offer storage, at varying costs : "free" storage often implies giving up privacy. In the end, many systems fail at keeping data private to the user while providing fine-grained sharing mechanisms.

In addition, the so-called "Web 2.0" services tend to hijack users' privacy by taking control of their information, and requiring people to trust them as the secret bearer. Recent trends in attacks, leaks and lawful spying prove that it is foolish to do so.

Storage providers in Tamias can not *mine* user data.

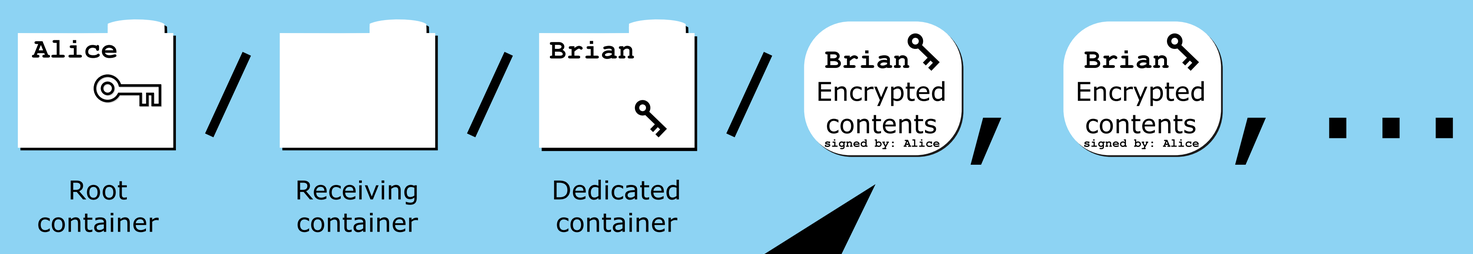
Public-key cryptography for trust management

- User **identity** is defined as a public/private **keypair**
- Each user has a list of public keys identifying other users he can **trust** (i.e. share data with)
- This data is stored within the system and encrypted by the owner's private key to keep safe



Directed "opt-in" sharing

- User Alice has to **grant** sharing privilege to user Brian in order **to receive** his links
- This requires user Alice to **create** a container for user Brian in her **private** space
- This container is for Alice to read links, and for Brian to write links



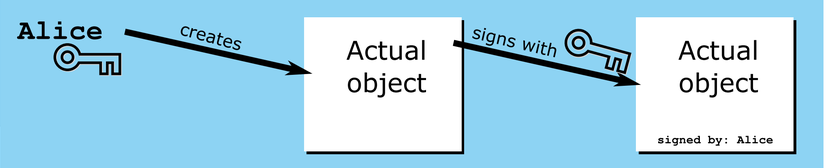
System architecture summary

Objects are stored in a distributed peer-to-peer network.
Stored objects can be shared with trusted persons.
Introduction is necessary before users can trust each other.
Trust can be revoked at any time, as well as sharing rights, or per-object access rights.

Each user builds a personal tree structure that is anchored at a secret root point. The identifier of this root point is derived uniquely and deterministically from the user's own private key.
The tree has special places for user's own files, links to shared files, writable shared zones, and readable shared zones.

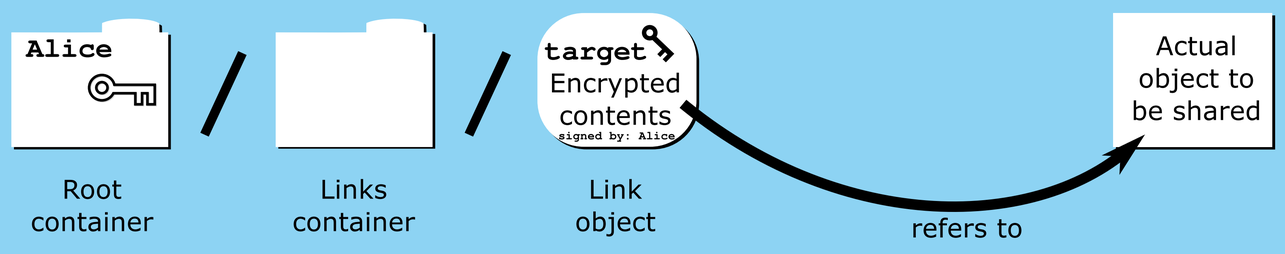
Data ownership

- Objects created within the storage are **signed by** the **owner**
- Link objects contain a **signed** access **authorization** for the intended target
- Storage **servers** can **validate** such **authorizations** using only local data



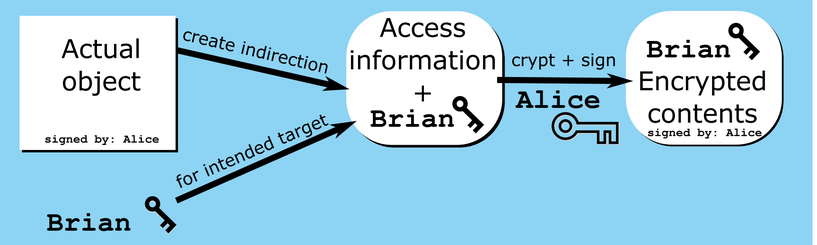
Sharing indirection

- Access is granted at object level via a **link** object intended to a unique recipient
- Link contents are **encrypted** using public-key of intended recipient
- Link objects are kept in a specific place of the root container



Link creation

- Links have a **source** and a **target**
- Whole bundle is **signed** by **source**
- Bundle is **encrypted** for **target**



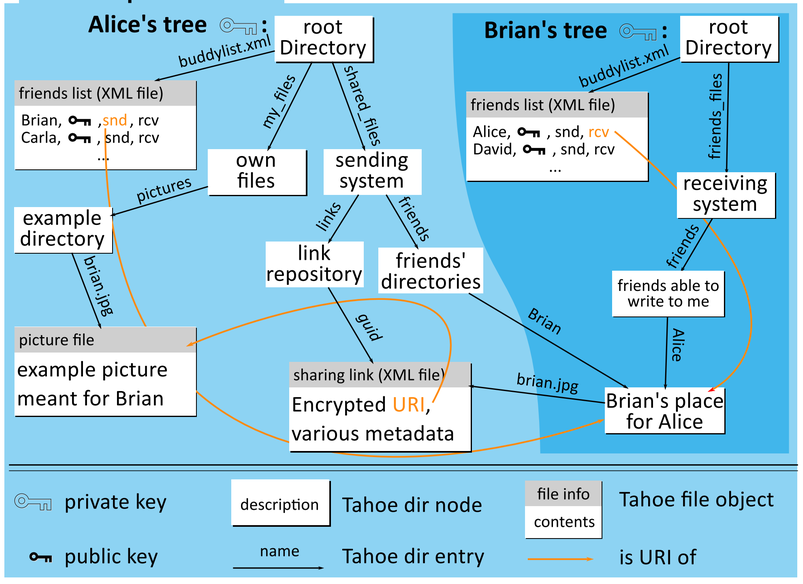
Tahoe-LAFS

To implement **TAMIAS** we are building on top of the **Tahoe-LAFS** system that has a few important features :

- distributed storage network
- full encryption of every object
- erasure coding for better reliability
- capabilities based access control

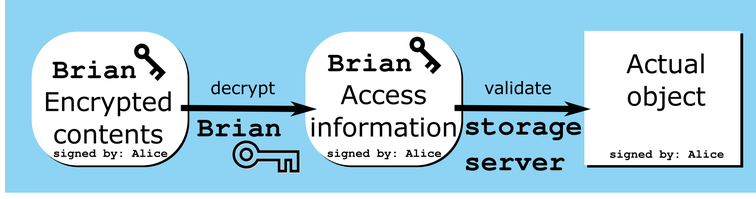
learn more on http://tahoe-lafs.org

example trees



Access control

- Client shows **owner-signed** bundle
- Client has to **match** bundle **target**
- Server **transmits** the required object



IIJ INNOVATION INSTITUTE

We would like to thank all the people that have provided feedback so far. We unfortunately do not have a public web page for this project at the moment but it will be announced on http://www.iij-ii.co.jp. You can also reach us by email : jean@iijlab.net