CDStore: Toward Reliable, Secure, and Cost-Efficient Cloud Storage via Convergent Dispersal

Introduction

CDStore builds on an augmented secret sharing scheme called convergent dispersal, which supports deduplication by using deterministic content-derived hashes as inputs to secret sharing. It combines convergent dispersal with two-stage deduplication to achieve both bandwidth and storage savings and be robust against side-channel attacks.

Publications

- Mingqiang Li, Chuan Qin, and Patrick P. C. Lee
  "CDStore: Toward Reliable, Secure, and Cost-Efficient Cloud Storage via Convergent Dispersal."
  Proceedings of USENIX Annual Technical Conference (ATC 2015), Santa Clara, CA, July 2015.

- Mingqiang Li, Chuan Qin, Jingwei Li, and Patrick P. C. Lee.
  "CDStore: Toward Reliable, Secure, and Cost-Efficient Cloud Storage via Convergent Dispersal."
  IEEE Internet Computing, 20(3), pp. 45–53, May-June 2016 (Special issue: Cloud Storage).
  (An earlier version appeared in USENIX ATC 2015)

Dependencies

CDStore is built on Ubuntu 12.04.3 LTS with gcc version 4.6.3. This software requires the following libraries:

- OpenSSL (https://www.openssl.org/source/openssl-1.0.2a.tar.gz)
- GF-Complete (https://github.com/ceph/gf-complete/archive/master.zip)
- boost C++ library (http://sourceforge.net/projects/boost/files/boost/1.58.0/boost_1_58_0.tar.gz)
- Leveldb (https://github.com/google/leveldb/archive/master.zip)

GF-Complete and Leveldb have been packed in `client/lib/` and `server/lib/` respectively.

Installation

Install OpenSSL, Boost C++ library and snappy (that is necessary for Leveldb) by the following command:

```
$ sudo apt-get install libssl1.0.0 libboost-all-dev libsnappy-dev
```

Type the following commands to compile GF_Complete in `client/lib/gf_complete`:

```
$ cd client/lib/gf_complete
$ ./configure
$ make
$ sudo make install
```

Configurations

After installing the necessary libraries, following the instructions to configure CDStore server and client.

Server

CDStore requires at least 4 servers for reliability. You need to configure and `make` them, respectively. In a successfully configured server, it has a directory `server/meta/` which stores the deduplication index, file recipes and share containers.

After a successful `make`, you will get the server program `SERVER`. You can start it by the following command, where `[port]` is the port that CDStore server serves in.

```
$ ./SERVER [port]
```

Client

Modify the configuration file `client/config` to specify the server information. For example, if you have run 4 servers with `./SERVER [port]` on machines:

```
0.0.0.0 with port 11030
0.0.0.0 with port 11031
0.0.0.0 with port 11032
0.0.0.0 with port 11033
```

You need to change `client/config` to:

```
0.0.0.0:11030
0.0.0.0:11031
0.0.0.0:11032
0.0.0.0:11033
```

Optionally, you can make advanced configure by modifying the total number of servers (4 by default), fault tolerance degree (1 by default), and security degree (0 by default) in `client/config`.

After all configurations, change directory to `client/` and type `make` to create client program `CLIENT`.

Quick Start

We provide scripts `auto_config.sh` and `auto_clean.sh` for quick configurations. You can run `./auto_config.sh` to compile and configure 4 servers, and run `./auto_clean.sh` to reset all configurations.

Usage

You can use the executable file `CLIENT` (in `client/`) in the following way:

```
usage: ./CLIENT [filename] [userID] [action] [secutiyType]

- [filename]: full path of the file;
- [userID]: user ID of current client;
- [action]: [-u] upload; [-d] download;
- [securityType]: [HIGH] AES-256 & SHA-256; [LOW] AES-128 & SHA-1
```

We show two usage examples:

```
// upload a file `test` from user 0 using high security mechanism (e.g., AES-256 & SHA-256)
$ ./CLIENT test 0 -u HIGH

// download a file `test`, from user 1 using low security mechanism (e.g., AES-128 & SHA-1)
// the downloaded file will be renamed to be test.d automatically
$ ./CLIENT test 1 -d LOW
```

Limitations & Known Bugs

- We assume the upload and download channels are secure (e.g., encrypted and authenticated), and do not implement mechanism for protection. We also assume that the user ID (input by CDStore client) is correct, so as to ensure the two-stage deduplication robust against side-channel attacks.

- When encode a test file generated via `urandom`, CDStore sometimes aborts with some errors like

  ```
  Error in region multiply operation.
  The source & destination pointers must be aligned with respect to each other along a 16 byte
   boundary.
  ```

  The error is possibly due to the library `gf_complete` that cannot encode random content.

Maintainers

- Current maintainer
  - Yanjing Ren, UESTC, tinoryj@gmail.com
  - JinGang Ma, UESTC, demon64523@gmail.com
- Original maintainer
  - Chuan Qin, CUHK, chintran27@gmail.com
  - Mingqiang Li, CUHK, mingqianglicn@gmail.com