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

Mingqiang Li, Chuan Qin, Patrick P. C. Lee
The Chinese University of Hong Kong

USENIX ATC’15
Multiple-Cloud Storage

Exploits diversity of multiple-cloud storage:

- Reliability
  - Fault tolerance
  - No vendor lock-in [Abu-Libdeh, SOCC’10]
- Security
Secret Sharing

- **Input:** secret; **output:** multiple shares

- **Properties:**
  - **Reliability:** secret is recoverable from enough shares
  - **Security:** secret is inaccessible without enough shares

- **Examples:**
  - Shamir’s [CACM’79]; Ramp’s [Crypto’84]; AONT-RS [FAST’11]
Challenges

- Secret sharing prohibits **deduplication**
  - Reason: Security builds on embedded randomness
    - Identical secrets lead to different shares
    - High bandwidth and storage overhead

- Our HotStorage’14 paper **convergent dispersal**:  
  - Replaces random input with deterministic hash derived from original secret

- **How to deploy in a real system?**
Our Contributions

- **CDStore**: a unified multi-cloud storage system with reliability, security, and cost efficiency
  - Also applicable for distributed storage systems

- A new instantiation of convergent dispersal
  - Higher throughput than our prior approach

- Two-stage deduplication
  - Bandwidth and storage savings
  - Secure

- Trace-driven experiments and cost analysis
CDStore Architecture

- Client-server model
- For whom? an organization that needs storage outsourcing for users’ data
- For what workload? backup and archival
Goals

- **Reliability:**
  - Availability if some clouds are operational
  - No metadata loss if CDStore clients fail

- **Security:**
  - Confidentiality (i.e., data is secret)
  - Integrity (i.e., data is uncorrupted)
  - Robust against side-channel attacks

- **Cost efficiency:**
  - Low storage cost via deduplication
  - Low VM computation and metadata overheads
Assumptions

- **Reliability:**
  - Efficient repair is not considered

- **Security:**
  - Secrets drawn from large message space, so brute-force attacks are infeasible [Bellare, Security’13]
  - Encrypted and authenticated client-server channels

- **Cost efficiency:**
  - No billing for communication between co-locating VMs and storage
Convergent AONT-RS (CAONT-RS)

- Extension of AONT-RS [Resch, FAST’11]
- Optimal asymmetric encryption padding (OAEP) AONT
  - Single encryption on a large block
- Other instantiations in our prior HotStorage’14 paper on Ramp’s and Rivest’s AONT

random key $\rightarrow$ secret’s hash

$n = 4, k = 3$
CAONT-RS Encoding

- Generate CAONT package \((Y, t)\):
  - \(h = H(X)\)
  - \(Y = X \oplus G(h)\)
  - \(G(h) = E(h, C)\)
  - \(t = h \oplus H(Y)\)

- Encode CAONT package with Reed-Solomon codes

**Code Notes**
- \(H(.)\): hash function (e.g., SHA-256)
- \(G(.)\): generator function
- \(E(.)\): encryption function (e.g., AES-256)
- \(C\): constant value block
Deduplication

Deduplication at the secret level
- Same secret → same shares that are dedup’ed
- Ensure the same share in the same cloud
  - Share $i$ stored in cloud $i$, where $i = 0, 1, ..., n-1$

Naïve approach: client-side global deduplication
- Saves most upload bandwidth and storage
- Susceptible to side-channel attacks
  - Attackers can infer if other users have stored same data
Two-Stage Deduplication

- Decomposes deduplication into two stages:
  - **Client-side intra-user deduplication**
    - Each CDStore client uploads unique shares of same user
    - Effective for backup workloads
  - **Server-side Inter-user deduplication**
    - Each CDStore server dedups same shares from different users
    - Effective if many users share similar data (e.g., VM images)

- Fingerprint index maintained by CDStore servers
CDStore Implementation

- C++ implementation on Linux

- Features:
  - Content-defined chunking (avg size = 8KB)
  - Parallelization of encoding and I/O operations
  - Batched network and storage I/Os

- Open issues:
  - Storage reclaim via garbage collection and compression
  - Multiple CDStore servers per cloud
  - Consistency due to concurrent updates
Experimental Setup

Testbeds:
- **Local machines**: Xeon 2.4GHz (slow), i5 3.4GHz (fast)
- **LAN**: Multiple i5 machines via 1Gb switch
- **Cloud**: Google, Azure, AWS and Rackspace

Datasets:
- Synthetic unique and fully duplicate data
- FSL dataset from Stony Brook University
  - Weekly file system snapshots
- Our own 156 VM images in a programming course
  - Weekly VM image snapshots
Encoding Speeds

- OAEP-based AONT brings high performance gain
  - CAONT-RS achieves 183MB/s on Local-i5

- Encoding speed slightly decreases with $n$
  - RS coding has small overhead

- Multi-threading boosts speed (details in paper)

Largest $k$ with $k/n \leq 3/4$
Storage Savings

- Intra-user dedup achieves high saving
  - At least 98% after Week 1

- Inter-user dedup is effective for VM dataset
  - Week 1: 93.4%
  - After Week 1: 11.8% - 47.0%
Transfer Speeds

- (Single-client) upload speeds in LAN:
  - Unique data ~ 77MB/s (network bound)
  - Duplicate data ~ 150MB/s (bounded by encoding + chunking)

- Performance in cloud bounded by Internet bandwidth

- Aggregate upload speeds increase with number of clients (details in paper)
Cost Analysis

- Compared to solutions w/o dedup:
  - (1) single cloud; (2) multiple clouds with AONT-RS
- At least 70% savings when dedup ratio is 10x – 50x
- Jagged curves due to switching cheapest VM instances
Conclusions

- **CDStore**: a unified multi-cloud storage system with three goals in mind: reliability, security, and cost efficiency

- **Building blocks:**
  - Convergent dispersal
  - Two-stage deduplication

- **Source code:**
  - [http://ansrlab.cse.cuhk.edu.hk/software/cdstore](http://ansrlab.cse.cuhk.edu.hk/software/cdstore)