Secure Deletion in the Swift Object Store

Tim Waizenegger, University of Stuttgart, Institute of Parallel and Distributed Systems

Problem Statement
- Secure, irrevocable data deletion is mandatory in many enterprise scenarios
- Current solutions rely on physical destruction of storage media
- With multi-tenant cloud storage systems, these solutions are no longer applicable
- Storage systems that use encryption still allow the reconstruction of deleted data

Cryptographic Deletion
- Use encryption to delete data:
  - Encrypted data can be stored in untrusted systems
  - Only keys need to be kept in a safe storage system where secure deletion is possible
  - Secure deletion of the keys translates to secure deletion of the data objects

Solution: Key Cascade

Properties of this key management schema:
- Supports a large amount of object keys
- Only a single master key must be kept in a trusted key store
- Secure deletion is possible on a per-object level with minimal overhead

How does it work?
- Group object-keys (okₙ) into partitions, encrypt those with partition keys (pkₙ)
- Encrypt the topmost partition with the master key
- Hashing on the object ID determines the partition and slot
- Tree height & partition size determine capacity and performance

How does it delete securely?
- Copy and re-encrypt partitions along the access path. Don't copy the object keys that should be deleted
- Finally create a new master key and securely delete the old one
- Without the old master key all old partitions can't be restored
  - This schema transfers the secure-delete capability of the master key store to all objects

References: https://www.ipvs.uni-stuttgart.de?id=Tim.Waizenegger