Lethe: Portable, Asynchronous Secure Deletion

Speaker: Eugene Chou (euchou@ucsc.edu)

Users have a legal right to securely delete their data stored on services that they use. Secure deletion requires that deleted data is irrecoverable, even if an attacker has physical access to the underlying device. Overwrite erasure is a traditional secure deletion technique; it erases data by overwriting in-place with random patterns. Unfortunately, overwrite erasure requires the ability to perform in-place updates, which is not supported by flash media, which is ubiquitous, and WORM media. This makes securely deleting data on storage solutions such as cloud incredibly difficult, due to the lack of knowledge of where data is placed and the storage media it is placed on, as well as the use of replication and versioning.

Cryptographic erasure is an alternative, efficient secure deletion technique; it encrypts user data with a cryptographic key before storing the data and erases the data by erasing the associated key. Fine-grained cryptographic erasure on data blocks places impractical storage requirements for naive cryptographic erasure; not only does each key need to be stored, each key must also be erasable. State-of-the-art secure delete systems address this with the technique of *large erasable storage*, which employs cryptographic erasure recursively in a tree hierarchy to reduce the required amount of key storage to a *single key*. Unfortunately, existing state-of-the-art secure delete systems suffer from high IO latencies due to their synchronous method of managing cryptographic keys and data to avoid data corruption. These existing secure delete systems are also inflexible since they manage encryption at the block layer, and cannot use the file system abstractions used by storage systems (e.g., cloud storage, network file systems, and FUSE storage systems).

We present Lethe, a portable and efficient secure delete system. Lethe uses large erasable storage to allow a user to securely delete data written by their applications, regardless of whether that data was stored on a local device, in the cloud, or even in a permanently public location. Lethe provides portability through its shim layer, which interposes on all file system operations performed by an application. The shim layer is designed to work with arbitrary *key management schemes*; schemes that manage cryptographic keys in a hierarchical manner. To simplify and accelerate crash-consistency operations, Lethe uses its key insight of *stable key management schemes*, which allows systems to perform efficient, asynchronous secure deletion.