Cryptographically Verified Design and Implementation of a Distributed Key Manager

Tolga Acar
Microsoft Research

Cédric Fournet
Microsoft Research

Dan Shumow
Microsoft Research

April 17, 2011

In this talk, we present DKM, a distributed key management system with a verified codebase. DKM implements a new data protection API. It manages keys and policies on behalf of groups of users that share the data. To ensure long-term protection, DKM supports cryptographic agility: algorithms, keys, and policies can evolve for protecting fresh data while preserving access to old data. DKM is written in C# and currently used by several large data-center applications.

To verify our design and implementation, we also write a lightweight reference implementation of DKM in F#. This code closes the gap between cryptographic models and production code. Formally, the F# code is a very precise model of DKM: we automatically verify its security against active adversaries, using F7, a refinement type-checker coupled with Z3, an SMT solver. To this end, we develop new libraries for cryptographic agility. Experimentally, the F# code closely mirrors the structure of our production code: we automatically test that the corresponding C# and F# fragments can be swapped without affecting the runtime behaviour of DKM. We discuss the computational soundness of our new cryptographic libraries. We also report on several problems we uncovered and fixed as part of this joint design, implementation, and verification process.