KISS: “Key it Simple and Secure”
Corporate Key Management

Zongwei Zhou, Jun Han, Yue-Hsun Lin,
Adrian Perrig, Virgil Gligor
ECE Department and CyLab,
Carnegie Mellon University

June 2013
Motivation

• Deployment of cryptographic systems and protocols (e.g., HTTPS) has grown rapidly
Motivation

• Key management is a **fundamental building block** of all cryptosystems

• Even experts fall prey to inadequate key management mechanisms
  
  – **DigiNotar CA**: keys are *misused* to issue certificates which enabled HTTPS man-in-the-middle attacks
  
  – **Stuxnet**: rogue device drivers were digitally signed by keys *stolen* from two high-tech companies
Challenges

• Fine-grained Key-Usage Control
  – Does an application executed by a user have permission to access a certain key?

• Secure System Administration
  – Communication between administrators and the Key Management System (KMS) must be authenticated
  – Stealing authentication credentials?
  – Insider attacks?
Existing Solutions

- **Hardware Security Module (HSMs)**
  - *Limited* control of key usage
Existing Solutions

- **Hardware Security Module (HSMs)**
  - **Limited** control of key usage
  - **Large TCB** for system administration
Existing Solutions

Software-only Solutions

• Deployment of KMS software on commodity servers

• Large TCB
  – Key protection, usage control and administration all rely on untrustworthy operating system services (e.g., process isolation, file system permissions)
System Goals

- Small and Simple TCB dedicated to KSM
- Cost-effective
- Secure System Bootstrap
- Secure System Administration
- Fine-grained Key Usage Control
Attacker Model

- **Malware** and **Malicious Administrators** attempt to leak, compromise, or misuse cryptographic keys.
System Design

KISS Server

KISS Client

Key Usage Control

Server Bootstrap

Remote Administration

Client Bootstrap

Trusted Admin Devices (TADs)

KISS Manager

Applications

10
Micro-Hypervisor Architecture

- Unified for server, client and manager

Untrusted Commodity OS

KISS Software

KISS Application

KISS Hypervisor

TAD

USB

NIC

Trusted Path

Trusted HW

Other KISS machines
Distinct Features

- Secure System Bootstrap
- Secure System Administration
- Fine-grained Key Usage Control
System Bootstrap

- Server bootstrap

Server Software

KISS Server

TPM

Extended Remote Attestation Protocol

Public Key

Private Key

TAD

TAD

TAD
Extended Remote Attestation

- TPM Quote includes KISS hypervisor, server software, server public key, TAD public key list
- Each TAD verifies:
  - Its own key is in the received TAD public key list
  - Length of the key list = # of TADs
- Minimum administrator effort
  - Checks that all TADs display success messages
- Security Analysis (e.g., Sybil attacks)
System Administration

- e.g., remote verification of server output

1. OUTPUT & Sig(OUTPUT) from KISS server
2. Manager display OUTPUT and Hash(OUTPUT) via trusted path
3. TAD verifies Sig(OUTPUT) using server public key, and display Hash(OUTPUT)
4. Admin uses TAD to remotely attest to KISS manager software and hypervisor
5. Admin confirms that two Hash(OUTPUT) match
System Administration

• Small and Simple TAD
  – Software: attestation, msg auth and bootstrap
  – Hardware: buttons, display ...
  – Usability: hash comparison
  – Used for local/remote and input/output
Key Usage Control

(2) KISS app is protected and verified by Hypervisor

(3) Client Software displays app information via trusted path for user confirmation

(4) **User** remotely attests to the Client Software and Hypervisor

(5) **User** authenticates to Client software

(1) **User** selects the KISS application to execute
Key Usage Control

• UserV helps defend against subtle attacks
  – e.g., stealing authentication credentials, or sensitive user input

• UserV is much simpler than TAD
  – Only performs remote attestation
  – Does not store any secrets
Conclusion

• A key management system architecture leveraging trusted computing techniques on commodity computers

• Small TCB: Micro-hypervisor-based design and lightweight administrator devices.

• Secure system bootstrap and administration, fine-grained key usage control
  – Defend against malware and insider attacks
Thanks!

zongweiz@andrew.cmu.edu