Squawk for Secure Devices

by Andrew Crouch

Outline

- **B**ackground
- **E**xample **P**roblem
- **T**he **P**rivilege **S**ystem
- **SKVM** vs. **S**quawk
- **F**uture **W**ork (for Squawk)
- **C**onclusion

Background

- What is a Secure Device?
  - If compromised (even physically) won't reveal secrets
  - Smart Card etc. …
- Current Access Privilege Mechanisms
  - Java privileges
  - Signed JAR files
  - J2se security architecture
- Squawk
  - Java VM written in Java
  - Suitable for small (embedded) devices

Example Problem

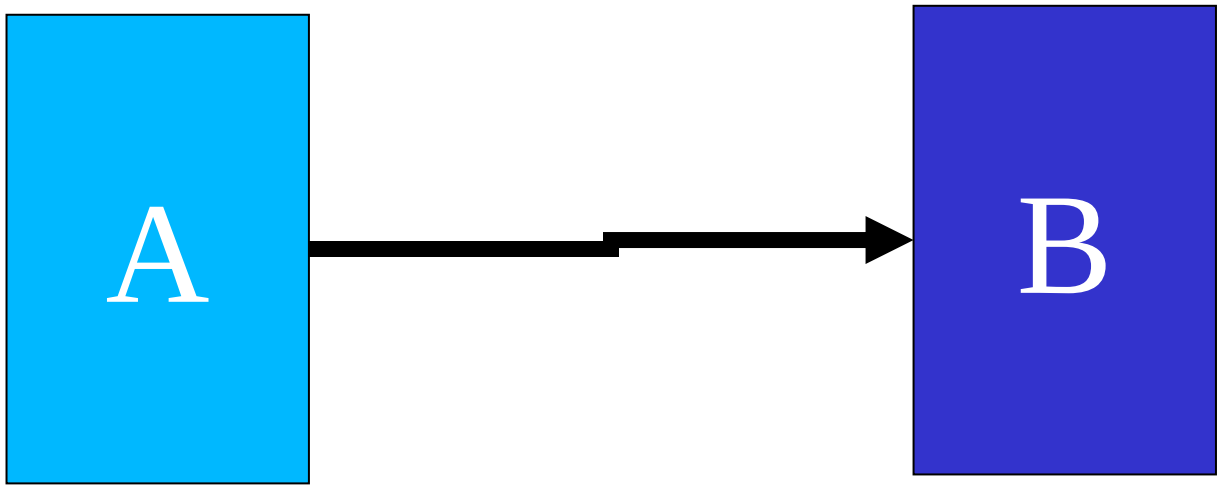
- Bank owns platform (creditcard/smartcard)
- Safeway approaches bank
- Safeway ‘points’ (loyalty program)

…

- Limit access to API, e.g.. I/O
- Java access privileges …
- Signed JAR …
- J2SE security architecture ..

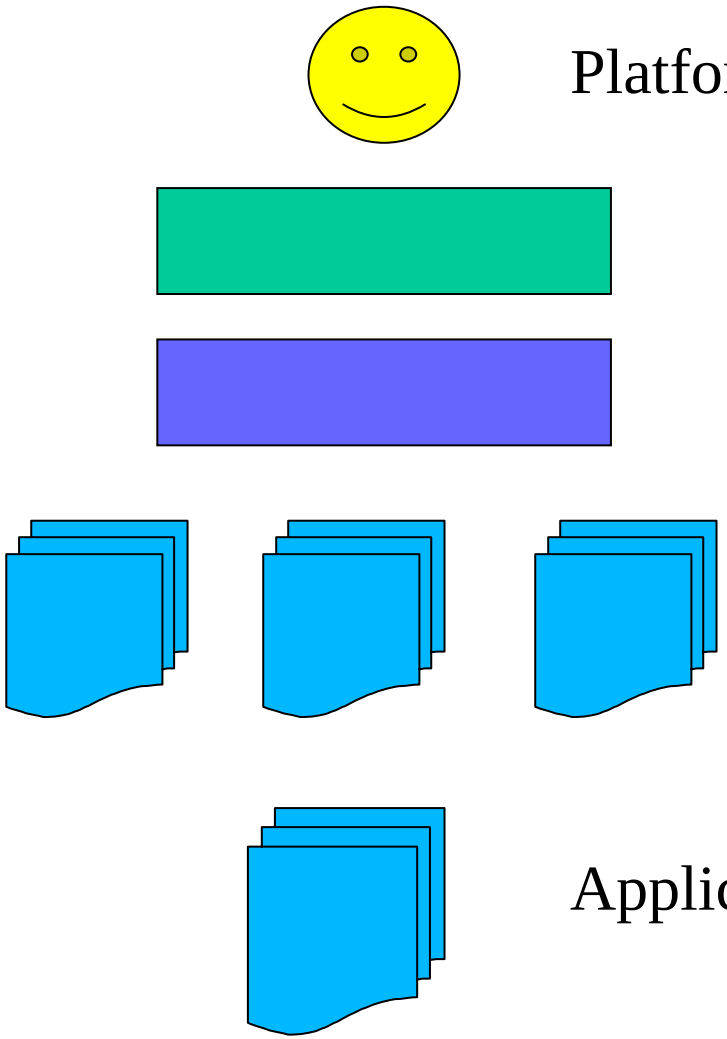
The Privilege System - I

- Per class privilege system
- Subclass, Class Resource Access & Domain



- For A to access B, A must have appropriate permission
  - I.e. call method, access field, instantiate or subclass

The Privilege System - II



- Chain of Trust,

-Rooted at Platform Owner

-Enforced by VM

The Privilege System - III

- Attached to each class
  - Subclass Key
  - Class Resource Access Key
  - Domain Key(s)
- Optional, depending on privileges
  - Privileges in form of Signed class file hash
  - Class Resource Access Permit
  - Subclass Permit
- Attached as extra “Attribute” in class file
  - Ignored by VMs that don’t understand
- Privileges verified when Loading and Linking

SKVM vs. Squawk

- Implementation exists, built on KVM, called SKVM
- SKVM written in C, ultimately Squawk entirely in Java
- Core procedures for privilege verification largely only porting effort.
- Cryptographic support in Squawk a minimal subset of java.security.*
- Test harness & test cases compatible/applicable to both

Future Work

- Modification of the mechanism is required for isolate migration and if application deployed as suite.
- Some form of performance metrics
- More user friendly tools for attaching privileges etc.
- Testing, testing, testing…

Conclusion

- Fairly heavy weight mechanism, possible issues for load times (eg. smart card)
- If deploy in object form (suites/isolates) overhead significantly reduced, but requires modification to mechanism
- Applications?

Questions?