Towards Permission-Based Attestation for the Android Platform

Ingo Bente

Trust@FHH Research Group
University of Applied Sciences and Arts in Hannover (FHH)

22 June 2011
Trust 2011
CMU Pittsburgh, PA

Trust@FHH
Fachhochschule Hannover
University of Applied Sciences and Arts

Agenda

Contents

Trust@FHH Research Group

Team

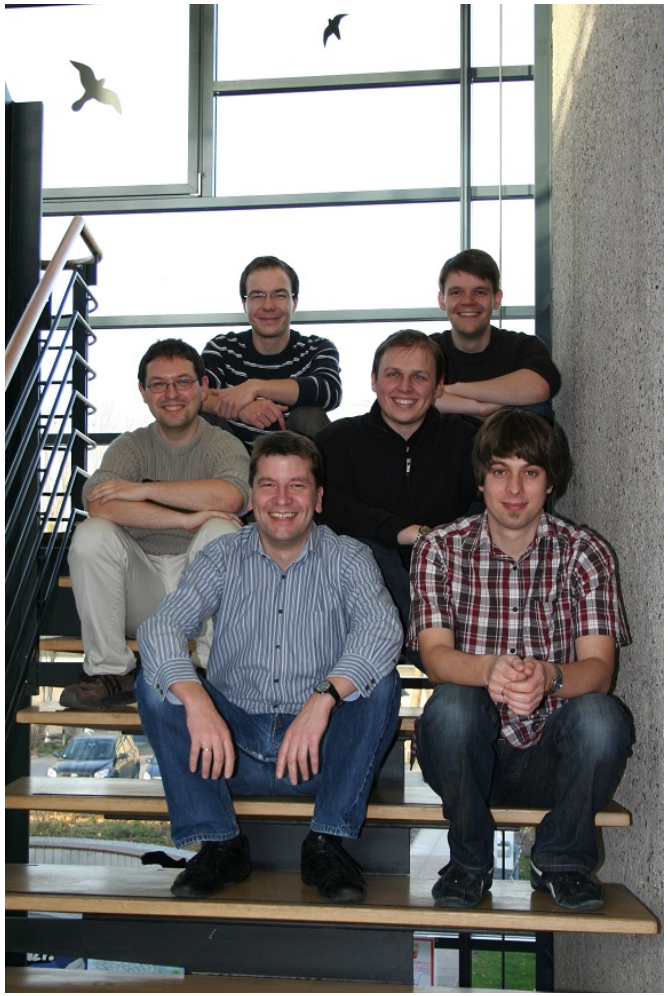
- head: Prof. Dr. Josef von Helden
- 3 research associates
- 4 student assistants

Research Fields

- Trusted Computing
- Network Security
- Mobile Security

More Information

- `trust.inform.fh-hannover.de`



Motivation

Mobile Malware

- malicious third party applications spreaded via "app stores"
- snoop for sensitive data (local phone data, sensors)
- abuse premium services (Trojan SMS)

Trusted Computing Concepts

- address malware issues in general
- binary remote attestation appropriate to counter malware threats

Binary Remote Attestation Drawbacks

- inherent issue: scalability
- lack of adoption (in general, not limited to mobile devices)

$\rightarrow$ **develop new attestation approach for mobile devices (Android)**

Idea of Permission-Based Attestation

Hybrid Approach

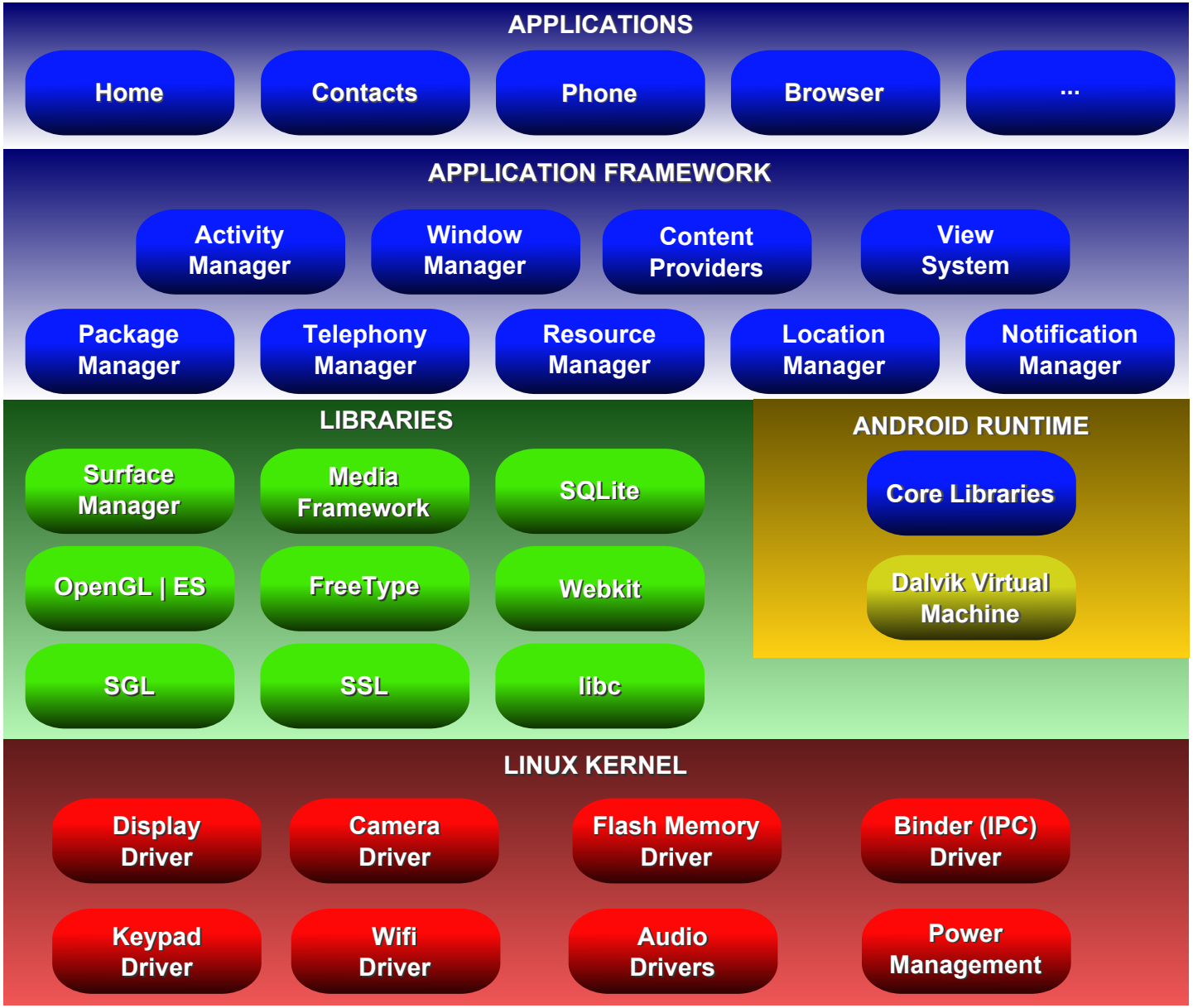
- general concept
  - binary attest only rather static part of the Android platform (excluding applications)
  - attest permissions used by applications (not their binaries!)
- $\rightarrow$ reduced complexity of chain of trust

Related Work

- Idea originated primarily from two prior approaches
- Kirin (Enck et al.)
  - security service for Android based upon permissions
  - third party apps are checked against predefined security rules
- Property Based Attestation (Sadeghi et al.)
  - attest security properties instead of application binaries
  - challenge: definition of reasonable properties

Contents

The Android Platform



Android Security Model

Isolation of Apps

- separate processes, separate file system
- each app is hosted by a dedicated Dalvik VM instance
- IPC via Binder API

Android Permissions

- permissions regulate access to phone resources
- apps list required permission in their manifest file
- primarily used in two ways
  1. permissions used by the app
  2. permissions to restrict access to the app's components itself
- Android platform enforces permissions

Example

- `ACCESS_FINE_LOCATION, INTERNET, RECEIVE_BOOT_COMPLETE`

Contents

Permission-Based Attestation Building Blocks

Static Chain of Trust (SCoT)

- binary measure before load components (extended to TPM)
- covers Android software stack (kernel, native libraries, Android runtime and application framework)
- apps are not included (exception see below ...)
- measurements are rendered to SML

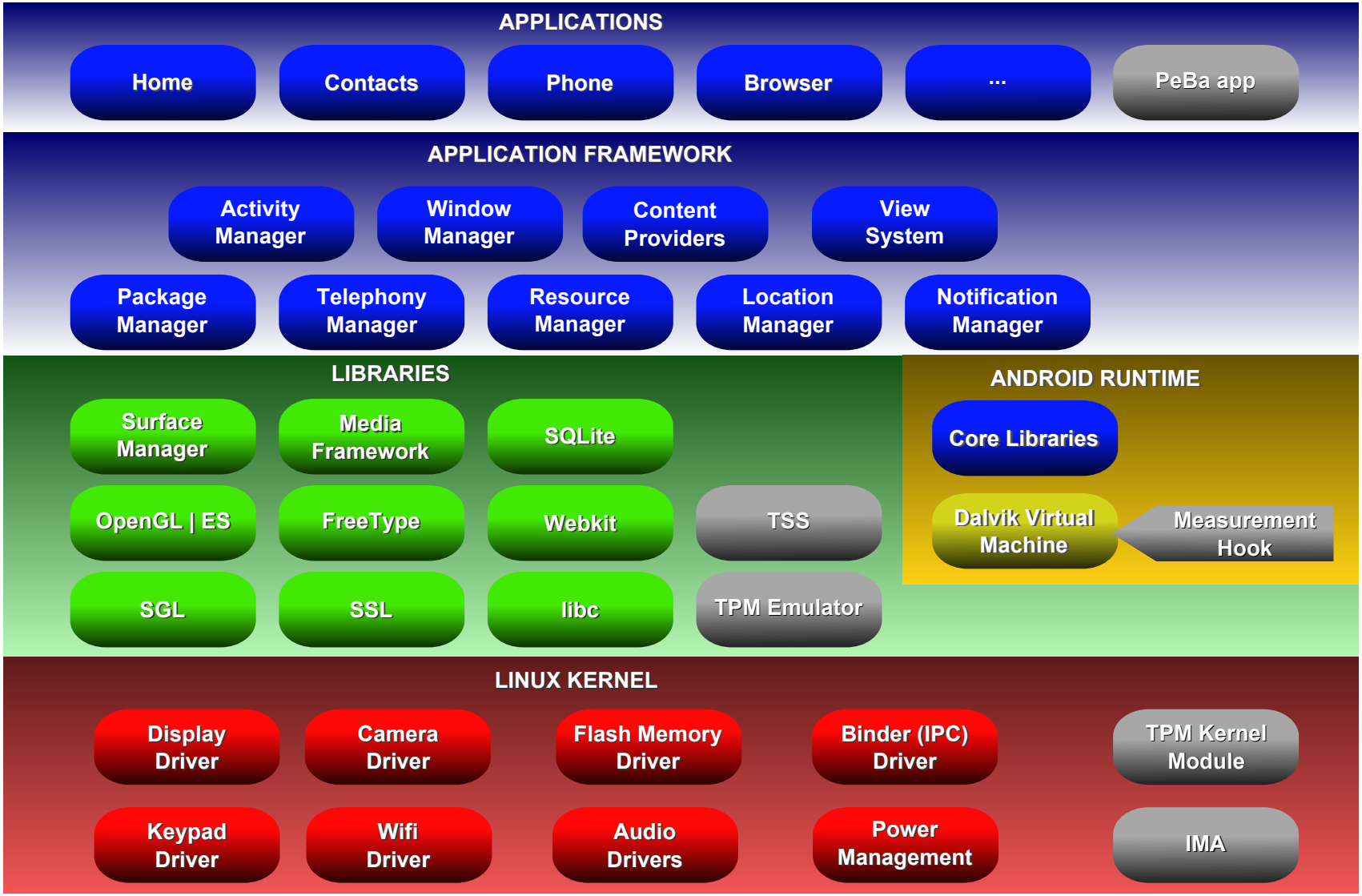
Permission-Based Attestation App

- the only app that is part of the SCoT
- measures requested permission labels of installed apps
- for each app extend TPM as follows: $PCR_n = SHA1(PCR_n \oplus SHA1(Permission_0 \oplus Permission_1 \oplus ... \oplus Permission_c))$
- maintains measurements in Permission Measurement Log (PML)

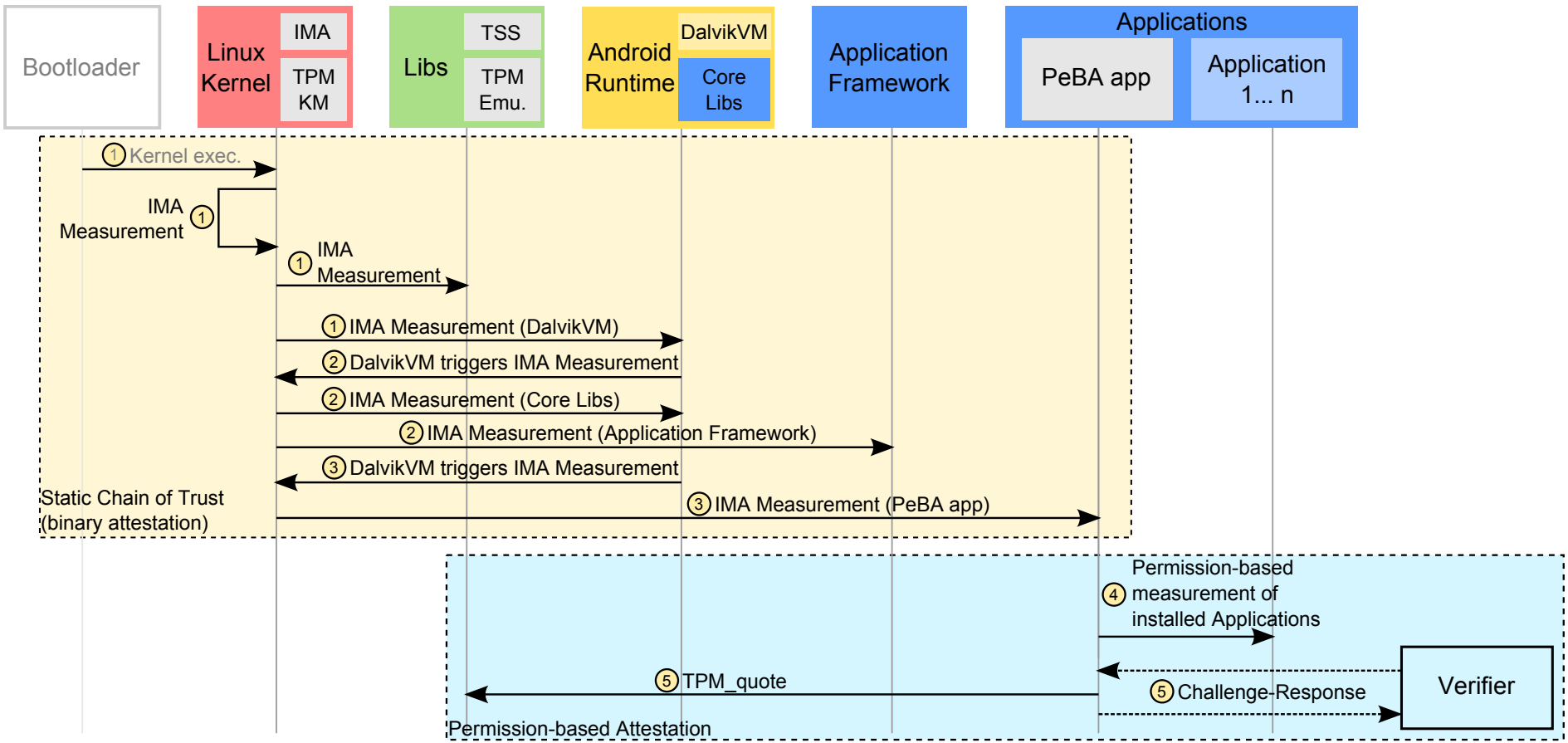
Permission Measurement Log Example

```
[...]
11 76f5ef2156db68c259d60b47280fbf156a054e2f com.android.contacts
        android.permission.CALL_PRIVILEGED
        android.permission.READ_CONTACTS
        android.permission.WRITE_CONTACTS
        android.permission.INTERNET
        android.permission.READ_PHONE_STATE
        android.permission.MODIFY_PHONE_STATE
        com.google.android.googleapps.permission.GOOGLE_AUTH.mail
        android.permission.WAKE_LOCK
        android.permission.WRITE_EXTERNAL_STORAGE
        android.permission.USE_CREDENTIALS
        android.permission.VIBRATE
11 6e4e78b206910d078f400ad061aa30d38562c146 com.android.phone
        android.permission.BROADCAST_STICKY
        android.permission.CALL_PHONE
        android.permission.CALL_PRIVILEGED
        android.permission.WRITE_SETTINGS
        android.permission.WRITE_SECURE_SETTINGS
        android.permission.READ_CONTACTS
        android.permission.WRITE_CONTACTS
        android.permission.SYSTEM_ALERT_WINDOW
        android.permission.INTERNAL_SYSTEM_WINDOW
        android.permission.ADD_SYSTEM_SERVICE
        android.permission.VIBRATE
[...]
```

Extended Android Platform



Flow of Operations



Contents

Limitations & Future Work

Prototype Limitations

- Android 2.2
- bootloader out of scope
- software TPM

Conceptual Limitations

- focus solely on statically requested permissions
- vulnerable to covert channels
- vulnerable to permission spreading

Future Work

- integration of further security policy details (intents)
- implementation of verifier

Thank You!