Analyzing ARM Native Code for Tracking Information Flow

Woo-Yeon Lee, Seo-Yoon Choi, Tae-Hun Kim, Byung-Gon Chun,
Cloud and Mobile System (CMS) Laboratory, Seoul National University (SNU)

1. Introduction

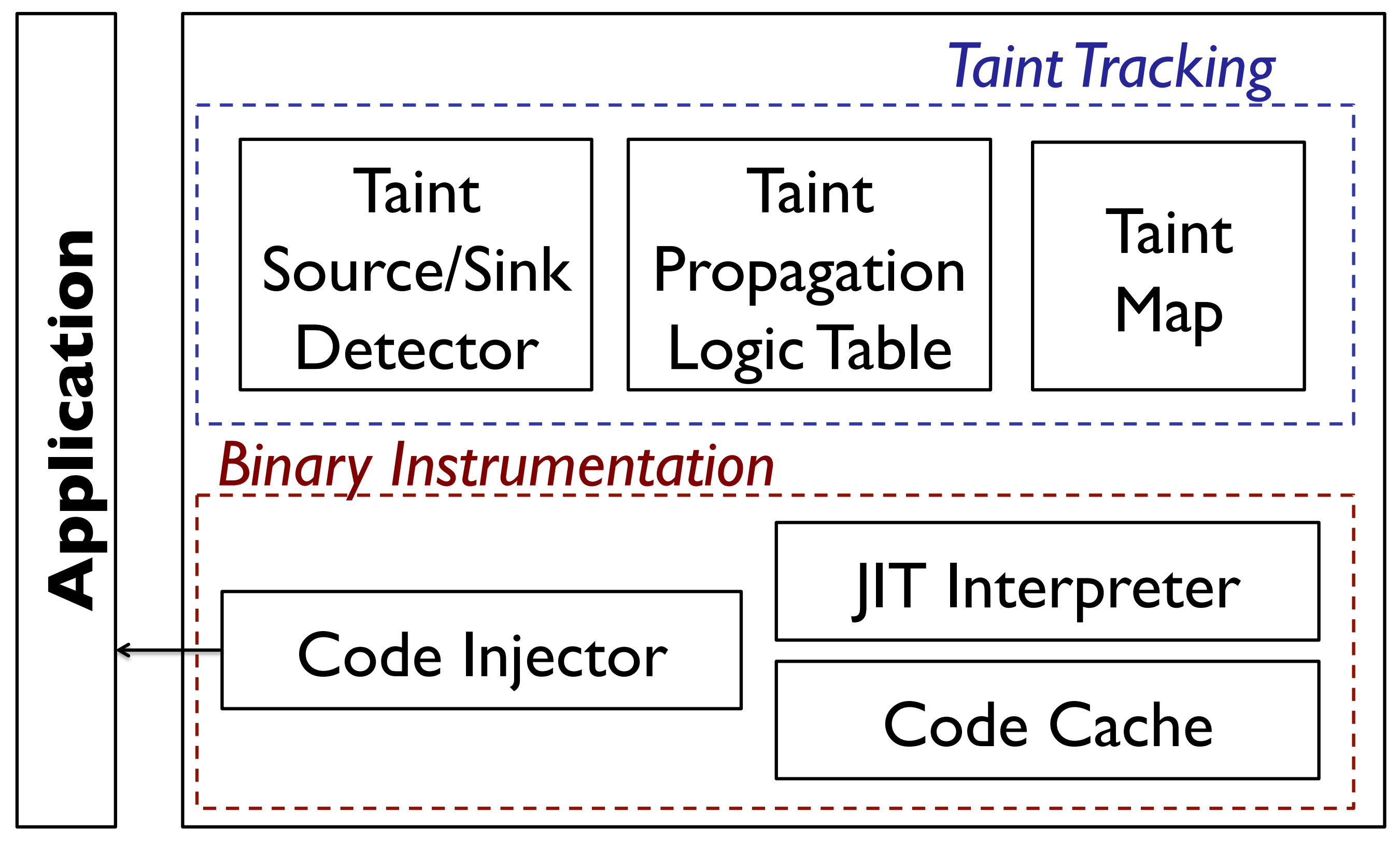
- Third-party "apps" may leak users' privacy-sensitive data or manifest malicious behavior.
- Why ARM native code?
  - More and more apps use ARM native code.
  - Android : 49% of the apps are packaged with third-party native library
  - Tizen : Native apps are written as ARM native code.
  - Lots of studies about information flow tracking, but not in ARM-instruction level.

2. ARM Architecture

- Advanced RISC architecture
  - 32bit-fixed instruction length
  - PC as a general register
  - Single execution cycle
  - Conditional execution
- Extension
  - Thumb / Thumb-2 mode (16bit)

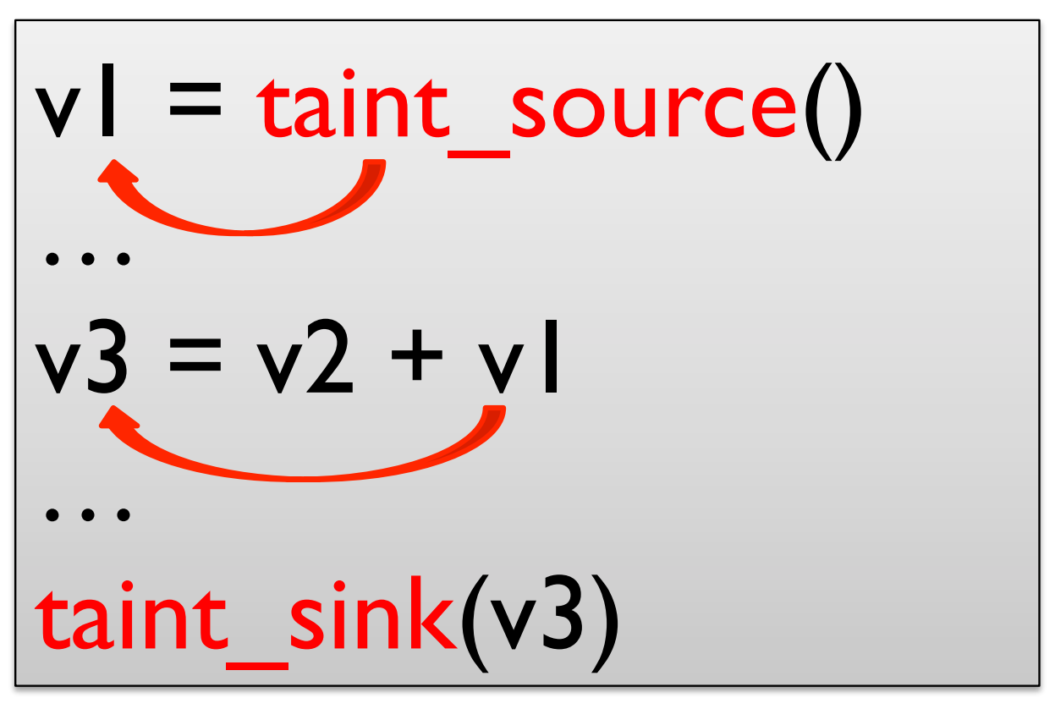
Some of these features are challenging to handle.

3. System Architecture

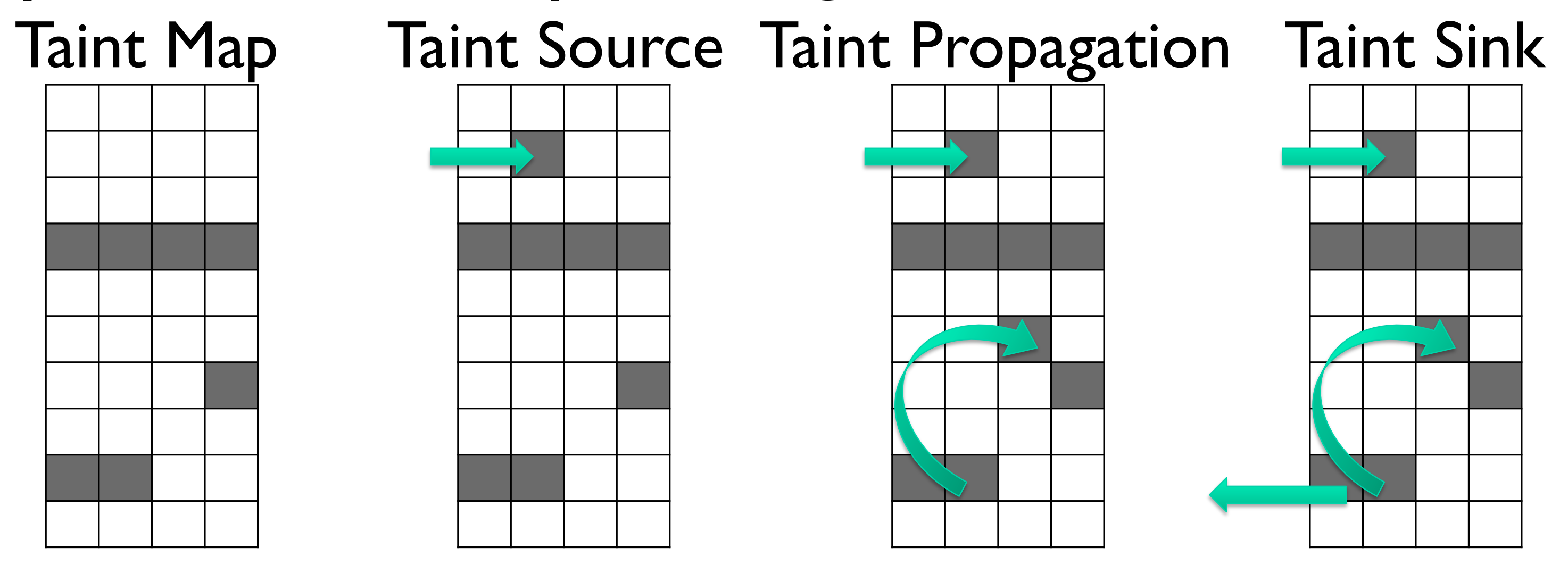


4. Dynamic Taint Tracking

- *Taint Tracking* is a technique used to track information dependencies from an origin
- Three Factors
  - Taint Source
  - Taint Propagation
  - Taint Sink



- Update Taint map during execution



- Taint Propagation Logic
  - We handle over 800 instructions

- Taint Map Function τ ( )

: τ *(A) retrieves the taint tag for 'A' from Taint Map.*

Operation Type	Assembly Representation	Action	Taint Propagation	Description
ADD <immediate>	ADD Rd, Rn, <immediate>	Rd := Rn + <immediate>	τ (Rd)$\leftarrow\tau$ (Rn)	Set Rd taint to Rn taint
ADD <register>	ADD Rd, Rn, Rm	Rd := Rn + Rm	τ (Rd)$\leftarrow\tau$ (Rn)$\cup\tau$ (Rm)	Set Rd taint to Rn taint OR Rm taint
MOV <immediate>	MOV Rd, <immediate>	Rd := <immediate>	τ (Rd)$\leftarrow\varnothing$	Clear Rd taint
MOV <register>	MOV Rd, Rn	Rd := Rn	τ (Rd)$\leftarrow\tau$ (Rn)	Set Rd taint to Rn taint

5. Dynamic Binary Instrumentation(DBI)

- What is *DBI?*

DBI manipulates executing binary at runtime and controls process's behavior.

Main challenge is an Application Transparency, keeping application behaving same as before instrumentation.

Before Instrumentation	After Instrumentation
ADD Rd, Rn, <immediate>	ADD Rd, Rn, <immediate> MOV τ (Rd), τ (Rn)
ADD Rd, Rn, Rm	ADD Rd, Rn, Rm OR τ (Rd), τ (Rn), τ (Rm)
MOV Rd, <immediate>	MOV Rd, <immediate> MOV τ (Rd), 0
MOV Rd, Rn	MOV Rd, Rn MOV τ (Rd), τ (Rn)

Taint Tracking with *DBI*,
- Low overhead enables real-time tracking

- Implementation
  - *Code Injector* inserts initial code into application to load analysis modules into application's memory space
  - *Taint Map* is implemented with *Shadow Memory*, which is consisted of shadow bytes mapping to bytes in main memory

CMS Labs Seoul National University