上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

ESORICS

On the (In)Security of Manufacturer-provided Remote Attestation Frameworks in Android

Ziyi Zhou[†], Xuangan Xiao[†], Tianxiao Hou[†], Yikun Hu[†]✉ and Dawu Gu [†]✉

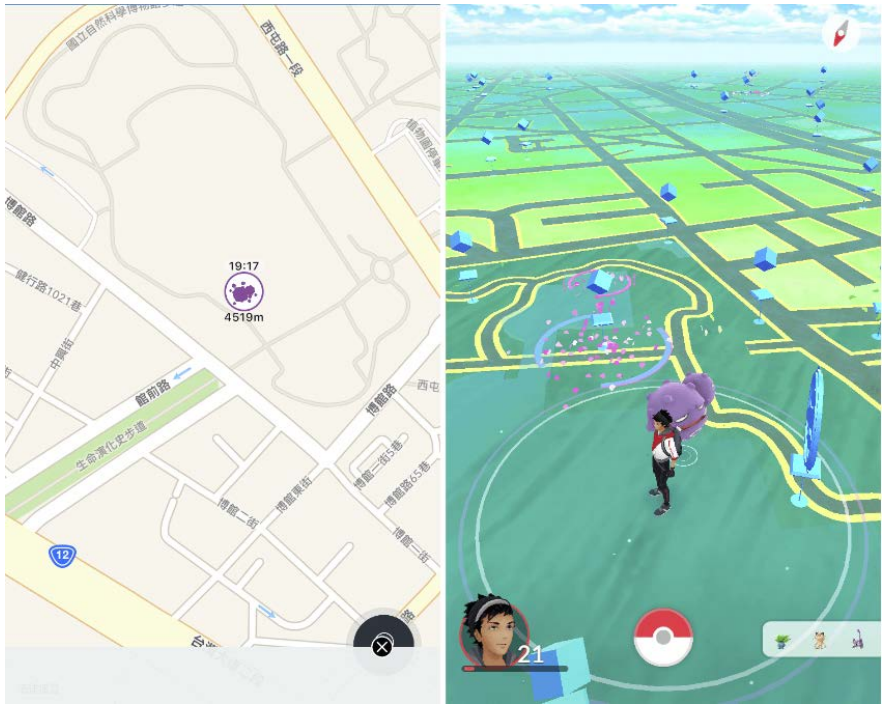
[†] Shanghai Jiao Tong University, Shanghai, China

Presented by Ziyi Zhou

September 27, 2023

1 Pokémon GO: A Case Study

- Over 572 million downloads[1]
- Over $6 billion in player spending[2]



1 *Pokémon GO Revenue and Usage Statistics (2023), https://www.businessofapps.com/data/pokemon-go-statistics/*

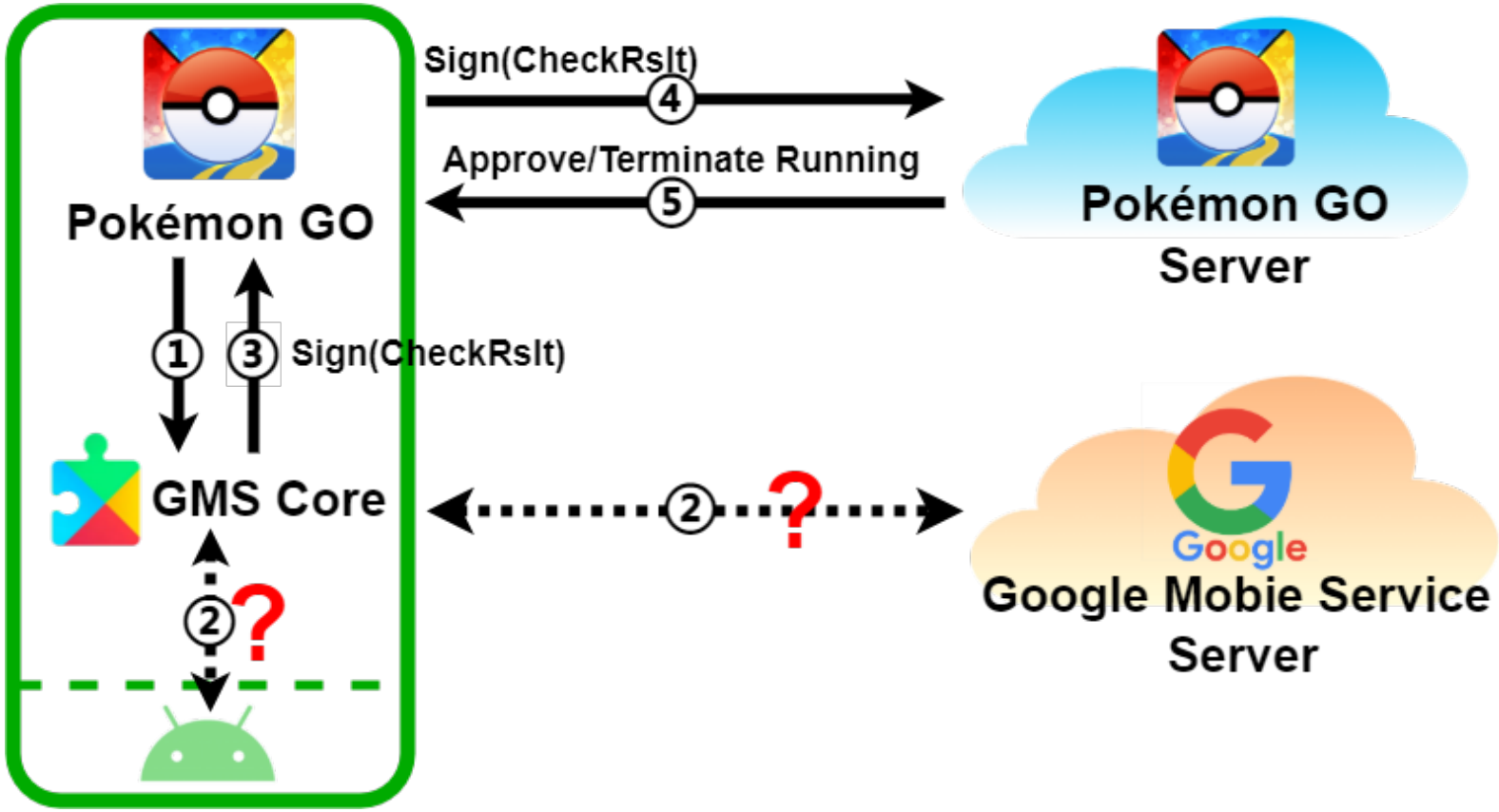
2 *Pokémon Go hits $6 billion in player spending, https://play.google.com/store/apps/details?id=com.nianticlabs.Pokemongo&hl=en_US.*

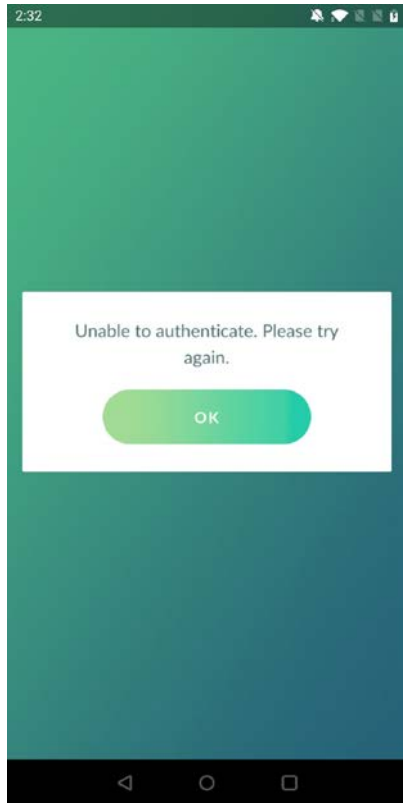
Mock locations instead of real locations?

- **Location spoofing tools**
  - **Type 1: do not require the device to be rooted**
    - E.g., FGL Pro, GPS JoyStick, Cha Cha Helper, Moloc
    - **Can be easily detected** (e.g., by checking the mock location checkbox)
  - **Type 2: require the device to be rooted**
    - E.g., Fake GPS Location Spoofer
    - **More difficult to detect**

How to detect/block location spoofing on rooted device (Type 2)?

- Pokémon GO has integrated **Google's SafetyNet Attestation** service





Manufacturer-provided Android Remote Attestation (MARA) frameworks

- **Google SafetyNet Attestation**

- **Huawei Safety Detect SysIntegrity**

- **What are the underlying mechanisms and protocols of these MARA implementations?**
- **Is there a generic way to bypass these MARA?**

Outline

Remote Attestation

- **Verify the integrity and trustworthiness of remote computing devices or systems**
  - **The Attester** generates information about itself ("Evidence")
  - **The Verifier** verifies the "Evidence" and generates the "Attestation Result"
  - **The Relying Party** makes the final decision based on the "Attestation Result" from the Verifier

- **In MARA:**
  - **The Attester:** mobile devices and third-party apps
  - **The Verifier:** the Manufacturer's Server
  - **The Relying Party:** the App Server

Mobile Service Core

- **MARA frameworks are often implemented in the MS Cores**
  - Google Mobile Service Core (GMS Core)
  - Huawei Mobile Service Core (HMS Core)
- **MS Cores are usually integrated into the OS**
  - Pre-installed since shipped from the factory
  - Installed by device users
- **MS Cores have over billions of users[1] and cover almost all countries[2]**
  - Google Play Store, YouTube, and Huawei Health

1 *Google I/O 2023: What's new in Google Play, https://io.google/2023/program/9019266d-186c-4a61-9cc5-b1c665eb40fb/.*
2 *HMS Core 5.0 launched for the global developers, https://www.huaweicentral.com/hms-core-5-0-launched-for-the-global-developers-comes-with-7-new-kits-and-services/.*

Integrity on Android

Device Integrity

- Rooting, Unlocking the bootloader, Changing the SELinux status, Using emulators, etc.
- Risks to users' property and privacy, Game cheating, Click Fraud, etc.

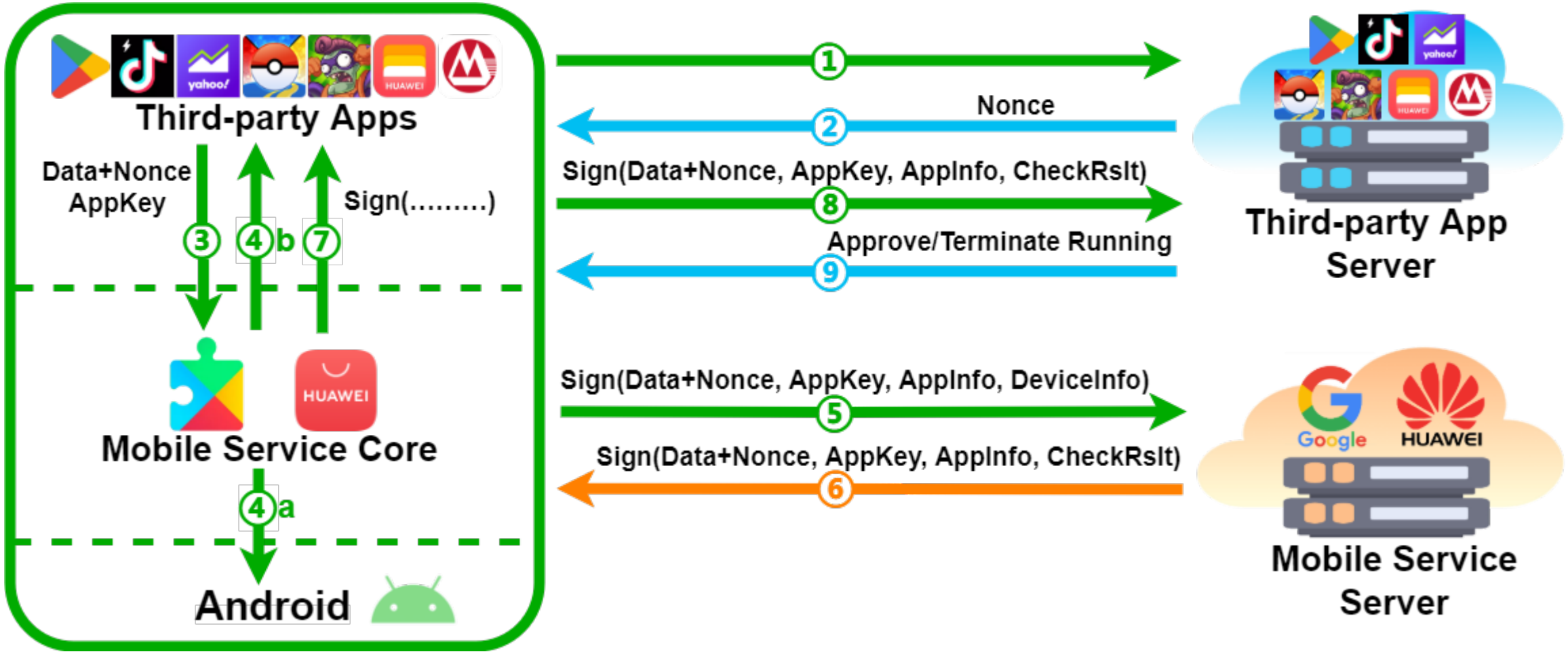
App Integrity

- App repackaging
- Intellectual property infringement, Ad insertion, etc.

Data Integrity

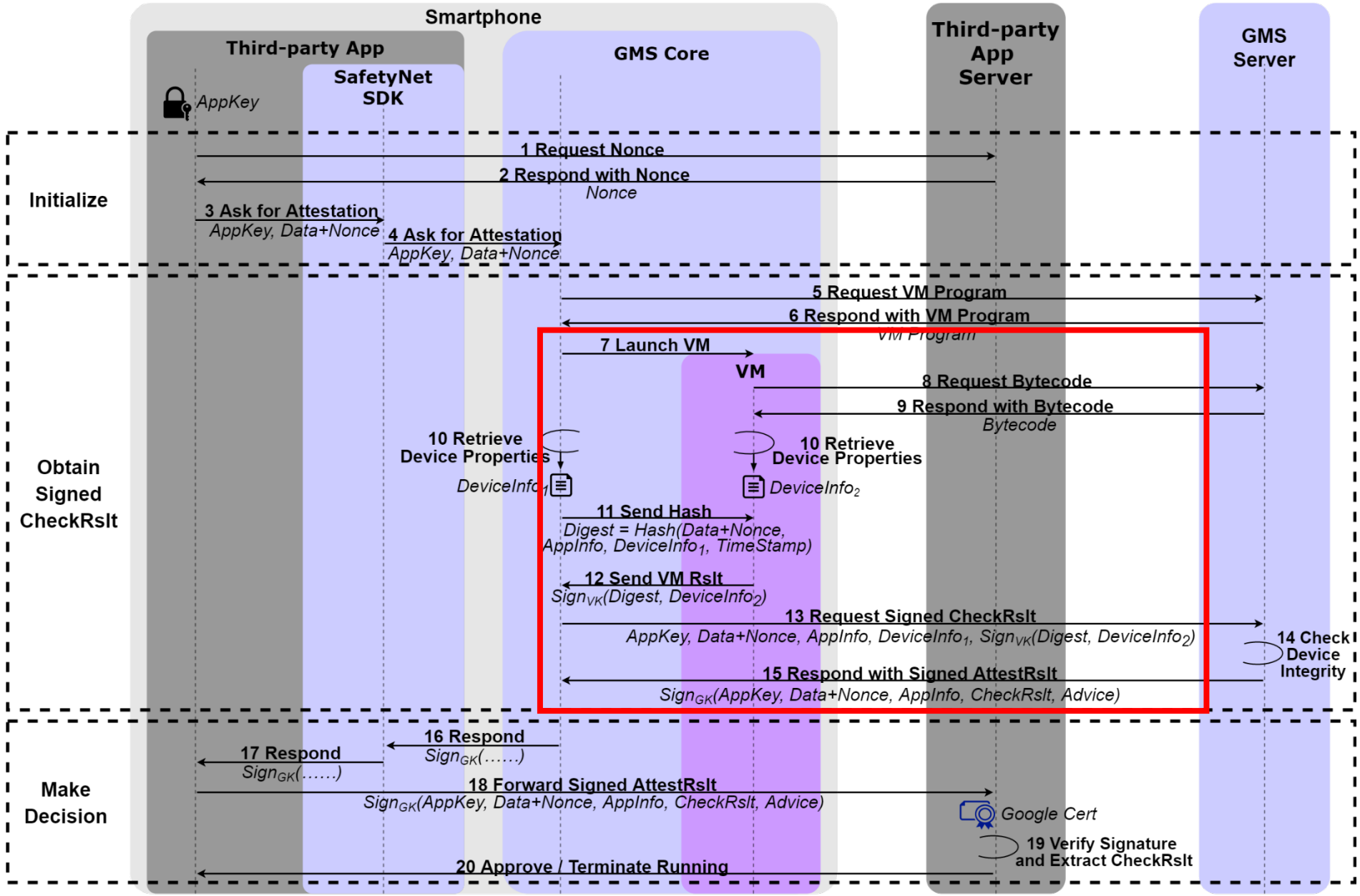
- Tamper with sensitive data through network MITM attacks

Through reverse engineering, we found that the attestation protocols of SafetyNet and Safety Detect followed a similar scheme.



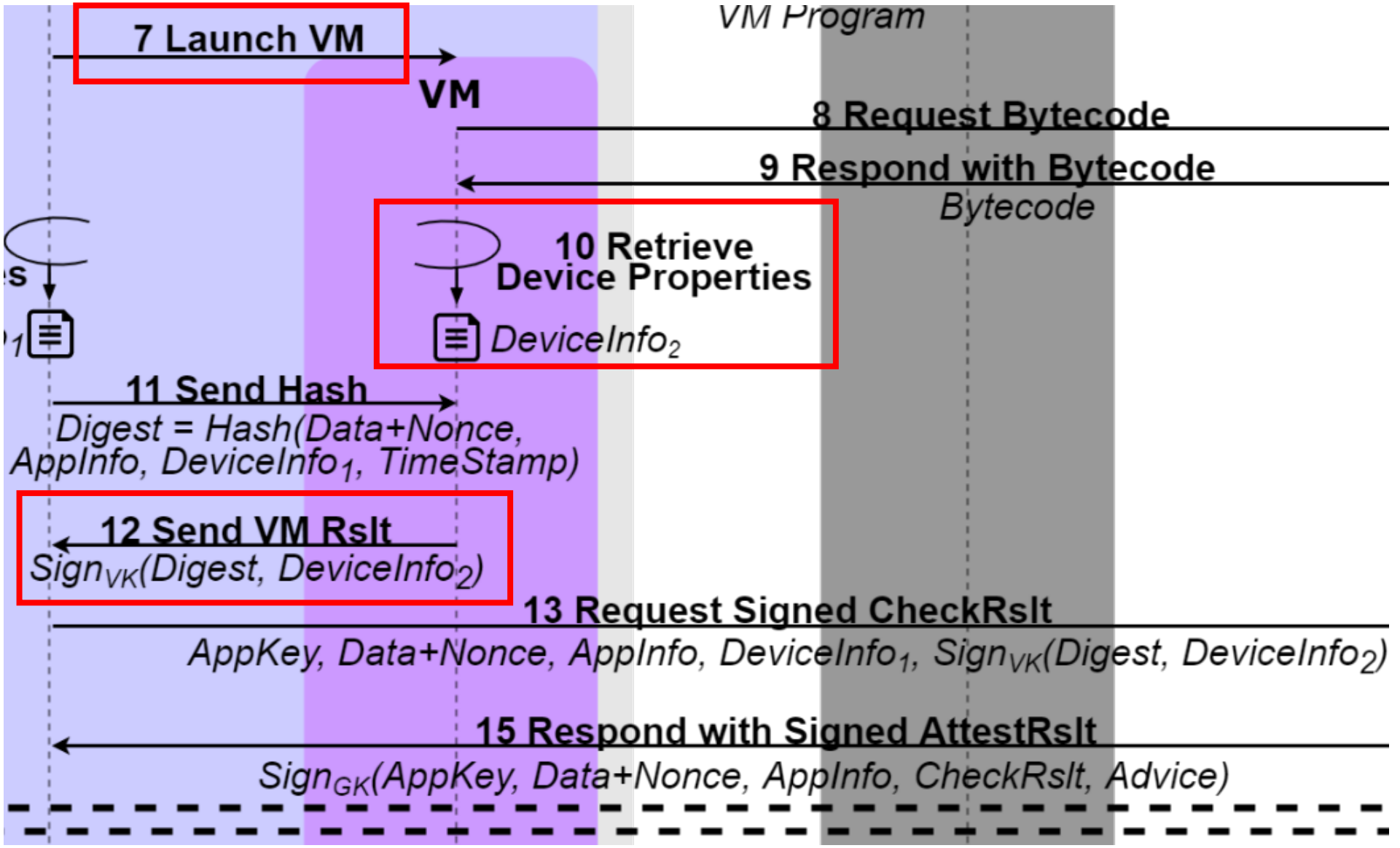
3 Demystifying MARA Frameworks

Details about SafetyNet



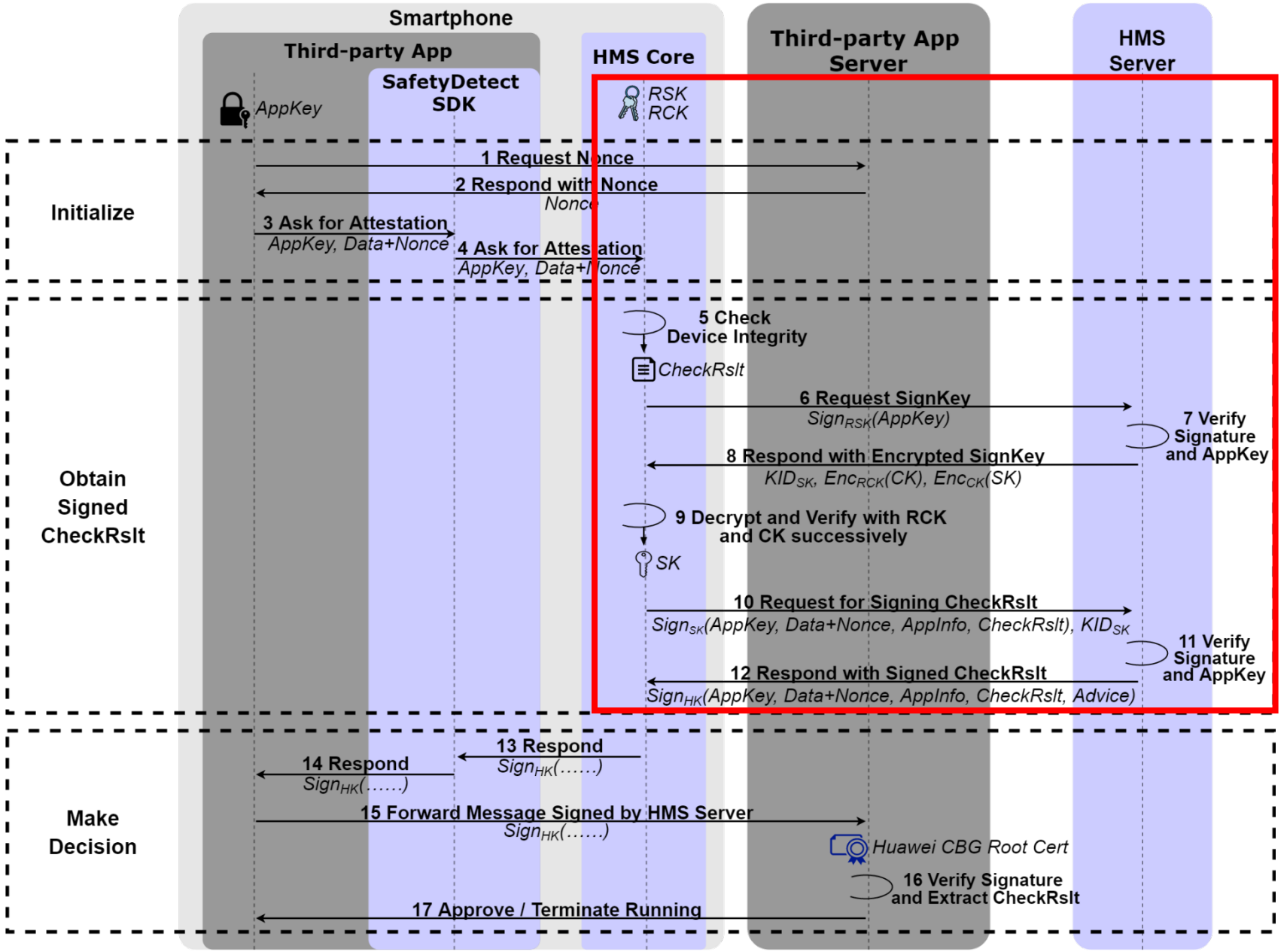
3 Demystifying MARA Frameworks

Details about SafetyNet

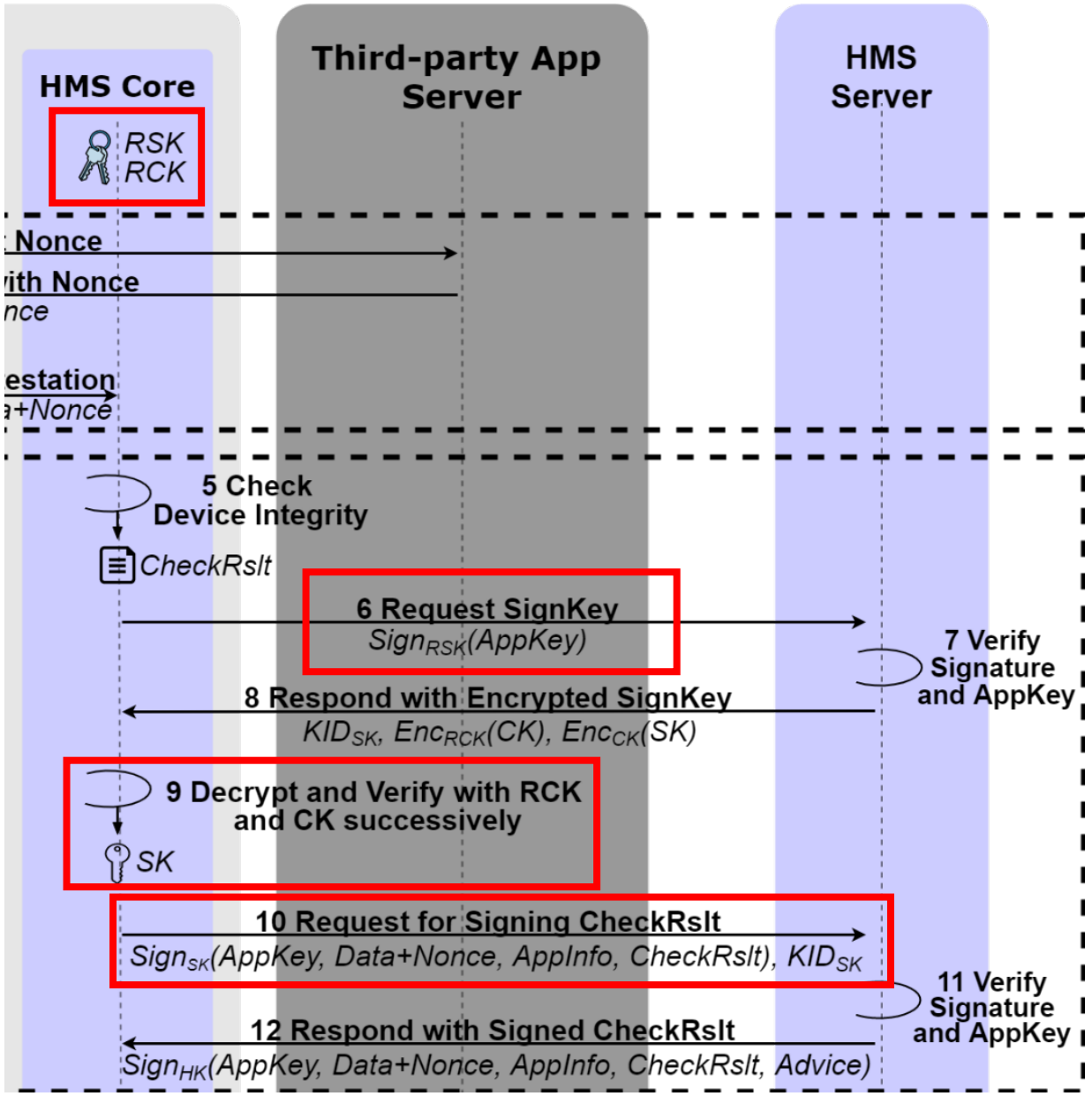


3 Demystifying MARA Frameworks

Details about Safety Detect



Details about Safety Detect



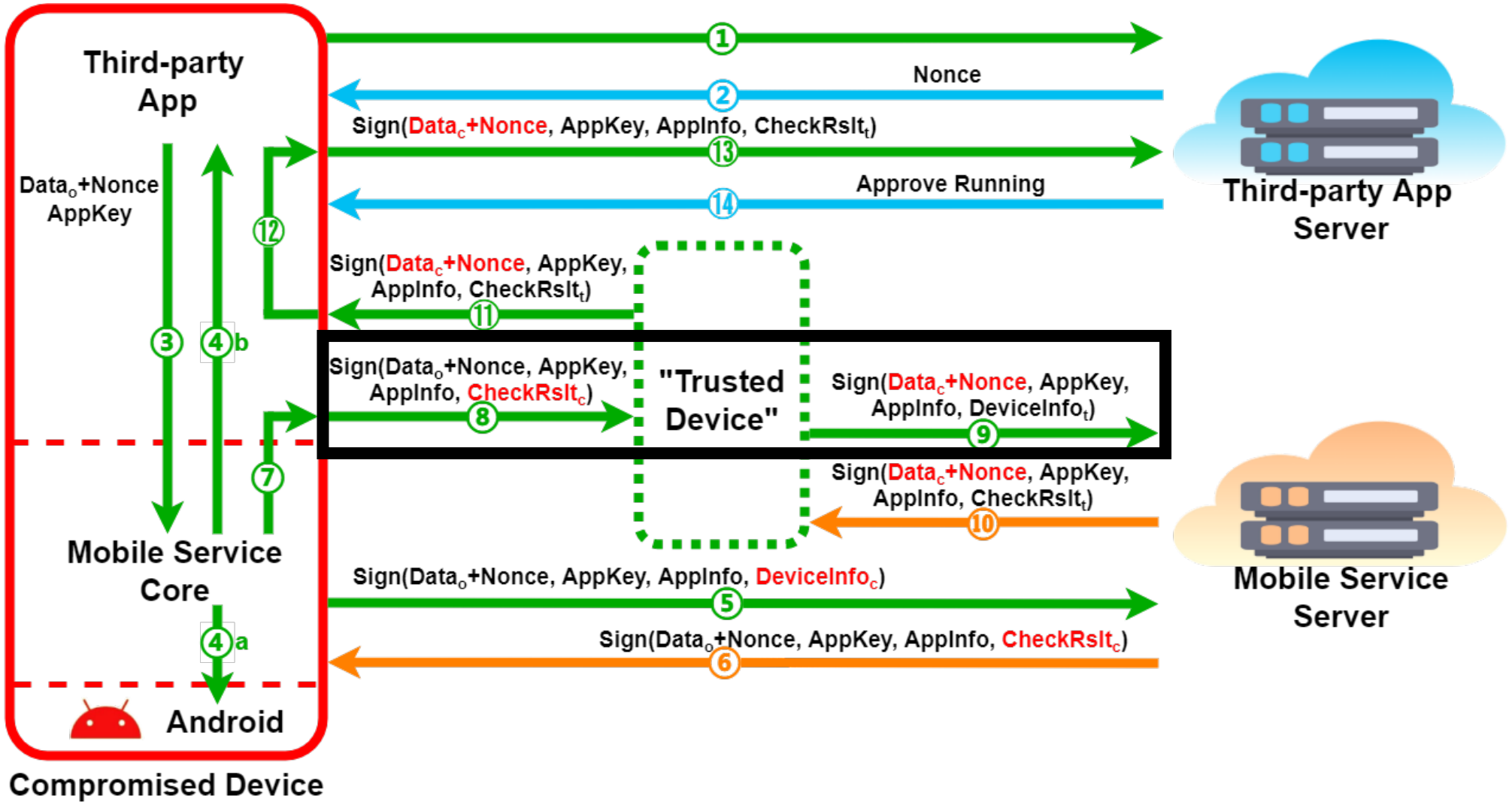
Scenarios and Attacker's Capabilities

- **Scenario 1: Bypassing Device Integrity Check**
  - An official app has been installed on a compromised device
  - *Attacker's Capability*: has root privilege
- **Scenario 2: Bypassing App Integrity Check**
  - A repackaged app has been installed on a non-rooted device
  - *Attacker's Capability*: can repackage the app
- **Scenario 3: Bypassing Data Integrity Protection**
  - Attacker hopes to manipulate HTTPS packets, and such operations usually require root privilege
  - *Attacker's Capability*: has root privilege

Fundamental Observation

- **Software-based DeviceInfo and hardware-based DeviceInfo**
  - Hardware-based part is only available for some devices
- **Most app servers do not have a mandatory requirement for hardware-based check result**
  - We conducted tests on over 35,000 popular apps
- **We can implement a "trusted device" to forge MS Core's signing process and launch a downgrade attack**

Bypassing Device Integrity Check & Data Integrity Protection



Attack Implementation

- **Implementation of "Trusted Device"**
  - **For HMS**: a protocol-emulating Python script
  - **For GMS**: need to support the running of the VM
    - We used Magisk to patch a ROM and flashed the ROM into a OnePlus 5T phone

- **Code Injection**
  - **Bypassing Device & Data Integrity Protection**
    - dynamic instrumentation with Frida
  - **Bypassing App Integrity Protection**
    - app repackaging with ShakaApktool

Effectiveness of Our Bypassing Approach

Details about the Android devices used

No	Model number	Android version	Build number	Bootloader status	GMS version	HMS version
#1	OnePlus 9R	11	Oxygen OS 11.2.4.4.LE28DA	unlocked	21.06.13	6.8.0.332
#2	Xiaomi Mi CC9 Pro	11	MIUI 13.0.4 Stable 13.0.4.0(RFDCNXM)	unlocked	21.21.16	6.8.0.332
#3	Oneplus 5T	10	H2OS 10.0.3	unlocked	22.12.15	6.8.0.332
#4	Nokia X5	9	00CN_2_15A_SP02	locked	22.12.15	6.8.0.332
#5	Xiaomi Mi 8	9	MIUI 10 9.8.22 Beta	unlocked	22.12.15	6.8.0.332
#6	OnePlus 5	9	H2OS 9.0.5	unlocked	22.12.15	6.8.0.332
#7	Motorola P30	8.1.0	ZUI 4.0.374 Stable	unlocked	20.12.16	6.8.0.332
#8	Xiaomi Mi 5	8.0.0	MIUI 10.8.11.22 Beta	unlocked	20.12.16	6.8.0.332
#9	Huawei Mate 9	7	EMUI 5.0	locked	10.2.98	6.10.4.300
#10	Lenovo K5 Note	5.1.1	VIBE UI V3.0	locked	10.0.84	6.8.0.332

Effectiveness of Our Bypassing Approach

Success rate of our bypassing approach compared with *Universal SafetyNet Fix* and *Shamiko*

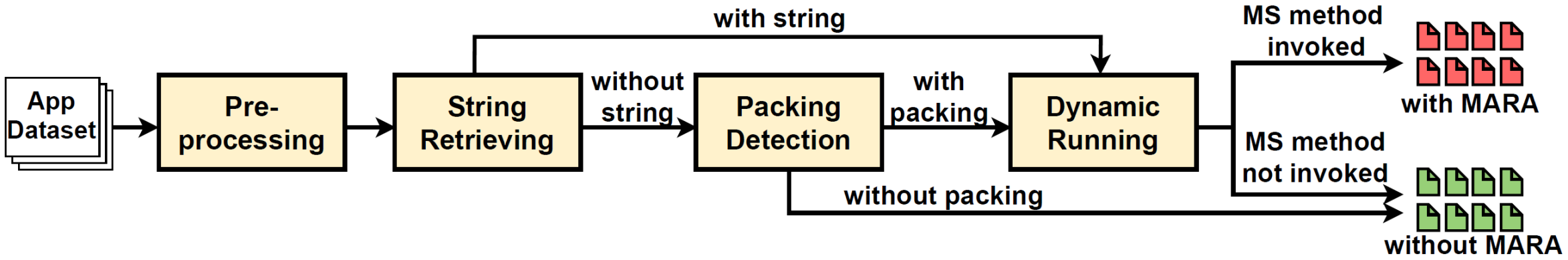
Byapsssing approach	Test item		Device No #1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Universial SafetyNet Fix	Device Int.	GMS	✓	–	✓	✗	✓	–	–	✗	–	–
		HMS	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗
	App Int.						✗					
Shamiko	Device Int.	GMS	✓	–	✓	✗	✓	–	–	✗	–	–
		HMS	✓	✓	✓	✓	✗	✓	✓	✗	✗	✗
	App Int.						✗					
our approach	Device Int.	GMS	✓	–	✓	✓	✓	–	–	✓	–	–
		HMS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	App Int.						✓					

Responsible Disclosure

- **For Google SafetyNet**: Google Security Team has filed the bug based on our report
- **For Huawei SafetyDetect**: CNCERT/CC has documented related vulnerability under CNVD-2023-57655

Large-Scale Measurement Study

Automated analysis pipeline for identifying affected apps



App measurement results

MARA	Total Apps	Static Analysis			Dynamic Analysis	
			String Retrieving	Packing Detection		
SafetyNet	35,245	potential	4,296	11,234	use	73
					don't use	11,161
		unsuspicious	30,949	24,011		
SafetyDetect	35,245	potential	226	7,158	use	31
					don't use	7,127
		unsuspicious	35,019	28,087		

Thanks for listening

Q&A