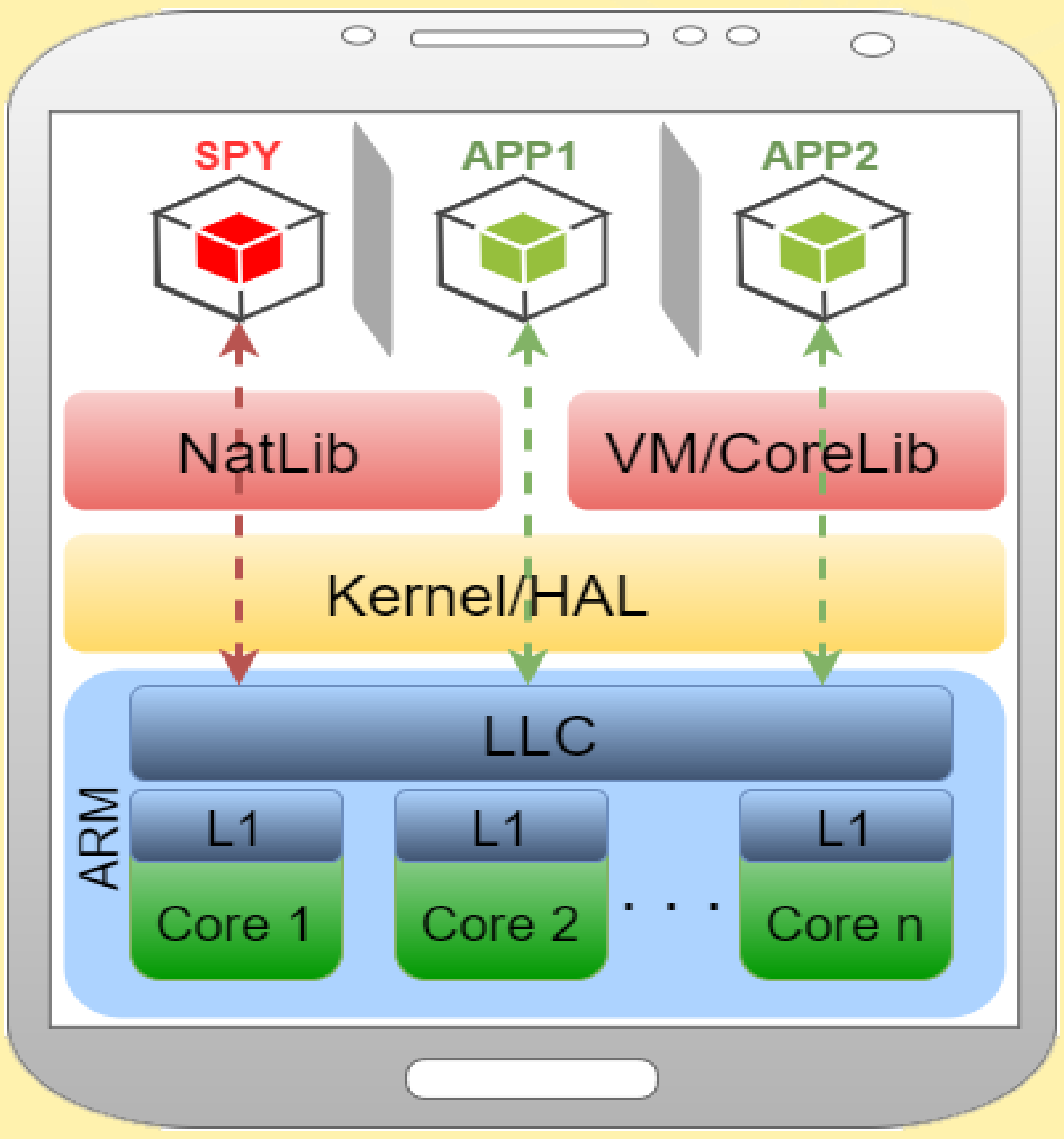
MIST: Systematic Analysis of Microarchitectural Information Leakage on Mobile Platforms

PIs: Thomas Eisenbarth, Berk Sunar
Worcester Polytechnic Institute, Vernam Group

WPI

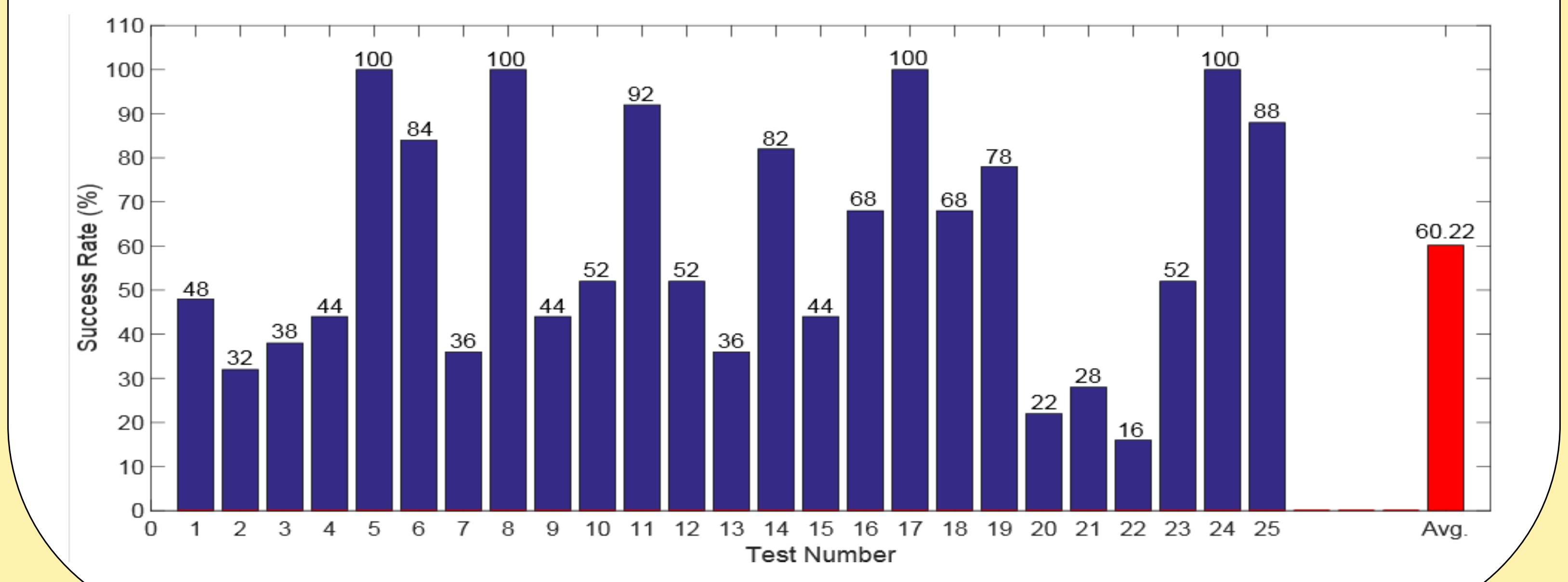
Mobile Computing

- Mobile computing is everywhere:
  Smartphones, Wearables, IoTs, Set-top Boxes…
  - Multiple 3rd party apps run on the same platform, sharing the underlying hardware [1]
  - Privacy and security sensitive information stored and processed on devices; passwords, mobile banking, health data, e-mails, photos
- Sandboxing is required and enforced by the operating system; Android and iOS [2,3]
  - Compute power and battery life are biggest challenges; require heavy optimizations



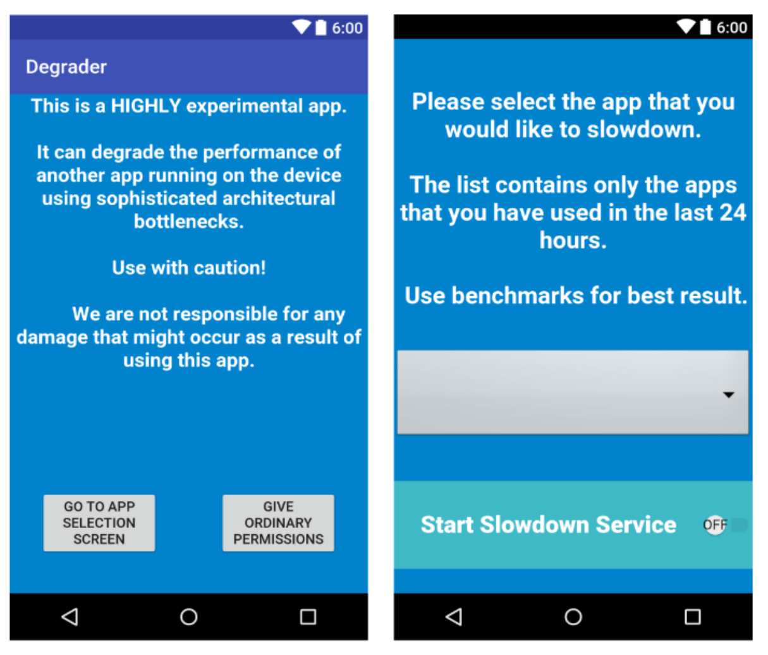
Machine Learning for Profiling

- Applying Machine Learning techniques to classify cache patterns of different benchmark apps on the cloud
- Detecting ping requests via the Last Level Cache reveals and identifies co-located VMs
- SVM-based classifier is trained on many different apps:
- The success rate of classifying 40 different apps is 60% on Amazon EC2, using cross-validation



Mobile QoS Attack

- Degrade the performance of a victim app up to 90%
- Background service detects when the victim app is in use
- Uses combination of microarchitectural features and logical channel leakages
- Is stealthy: CPU load of the attacker does not exceed 10%
- Currently neither detected by Google Play Store nor the top malware scanners in the market



This is a HIGHLY experimental app.

It can degrade the performance of another app running on the device using sophisticated architectural bottlenecks.

Use with caution!

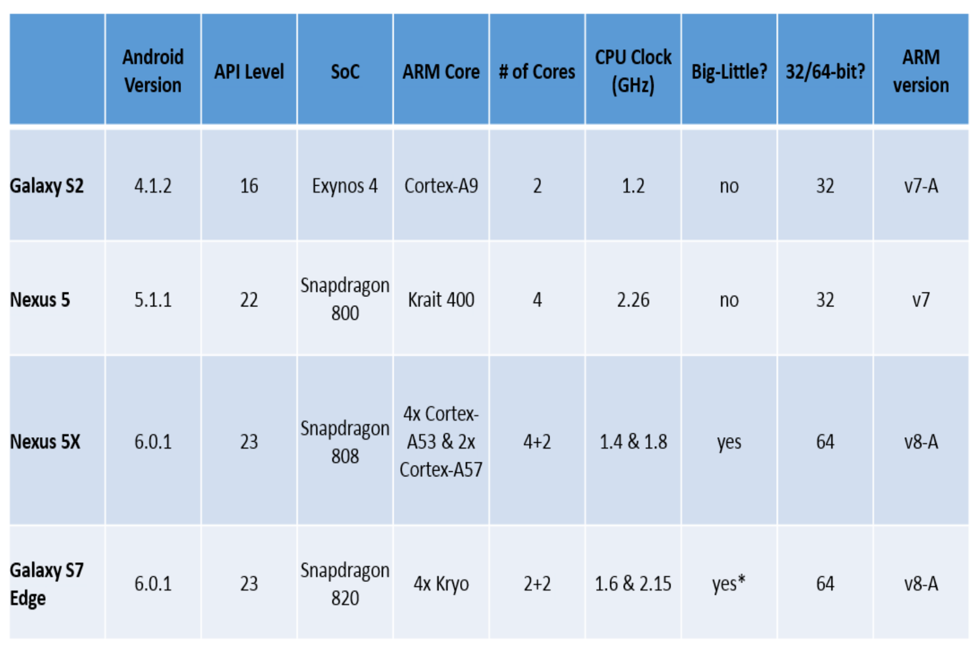
We are not responsible for any damage that might occur as a result of using this app.

Please select the app that you would like to slowdown.

The list contains only the apps that you have used in the last 24 hours.

Use benchmarks for best result.

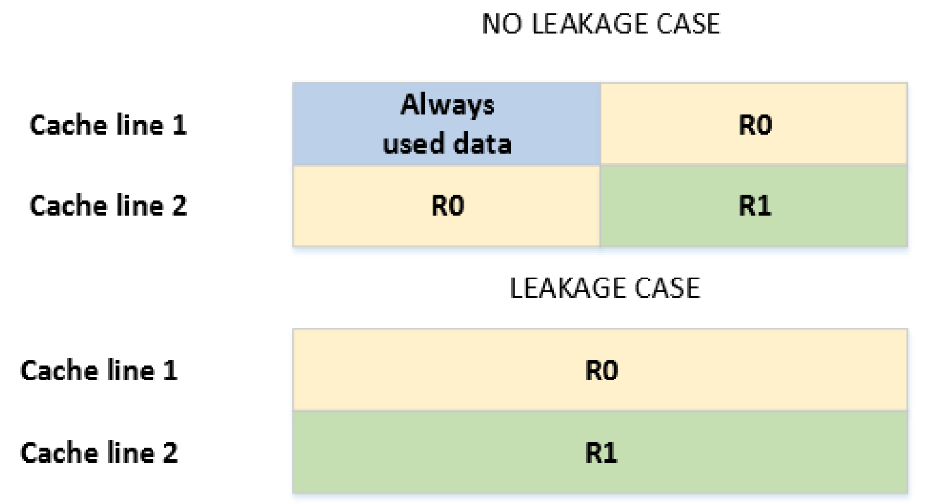
Start Slowdown Service
OFF

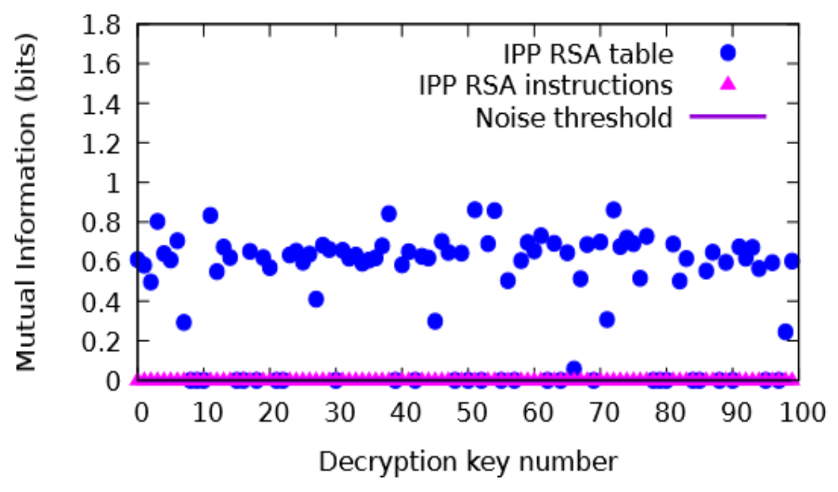


	Android Version	API Level	SoC	ARM Core	# of Cores	CPU Clock (GHz)	Big-Little?	32/64-bit?	ARM version
Galaxy S2	4.1.2	16	Exynos 4	Cortex-A9	2	1.2	no	32	v7-A
Nexus 5	5.1.1	22	Snapdragon 800	Krait 400	4	2.26	no	32	v7
Nexus 5X	6.0.1	23	Snapdragon 808	4x Cortex-A53 & 2x Cortex-A57	4+2	1.4 & 1.8	yes	64	v8-A
Galaxy S7 Edge	6.0.1	23	Snapdragon 820	4x Kryo	2+2	1.6 & 2.15	yes*	64	v8-A

Crypto Library Primitives

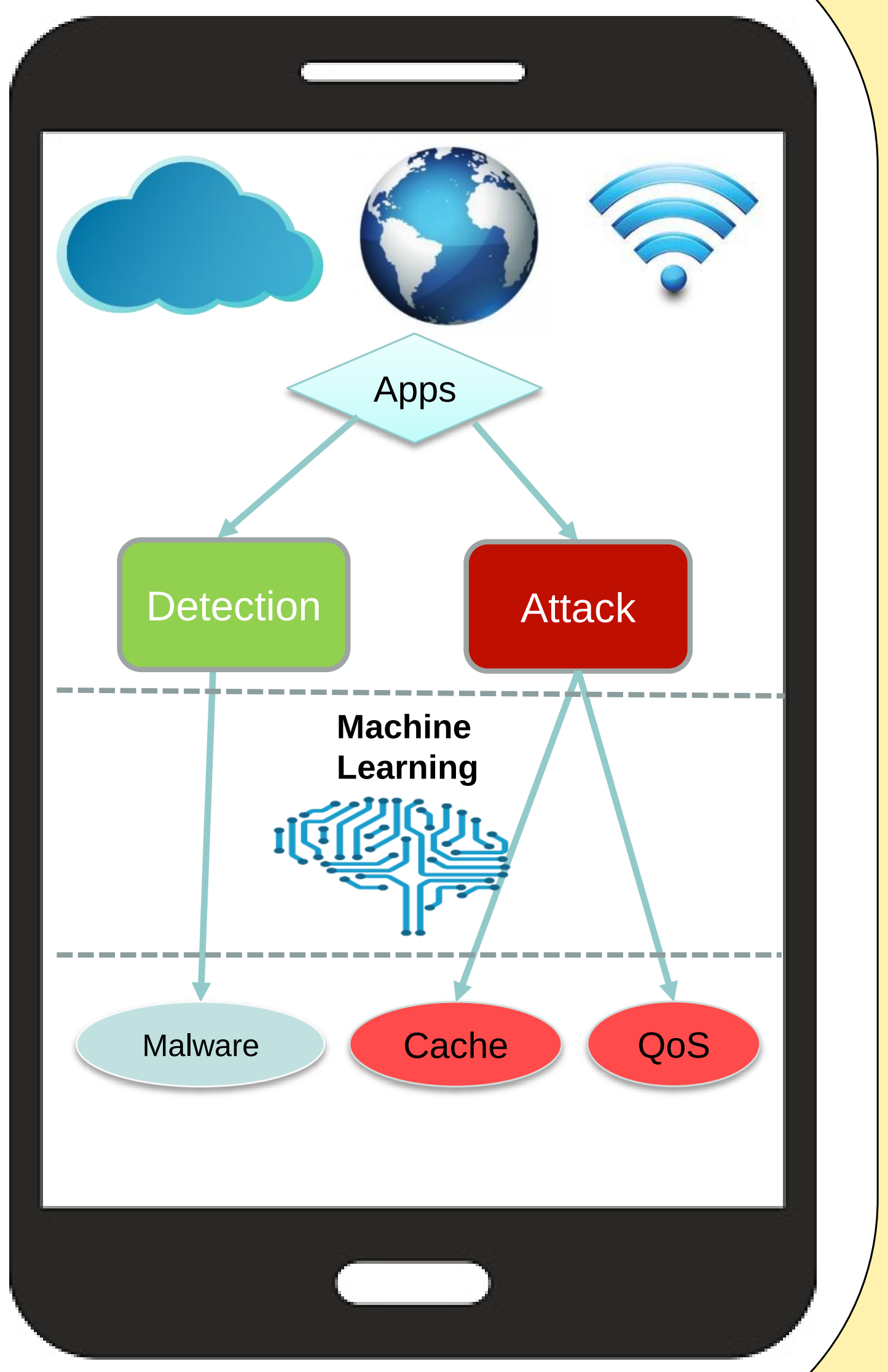
- Microarchitectural attacks exploit code design mistakes
- Security critical code should not feature secret dependent execution flow and memory accesses
- Creation of a methodology using cache traces and mutual information to validate the sanity of cryptographic code
- Evaluated the sanity of 8 well known cryptographic libraries (e.g WolfSSl, OpenSSl or Intel IPP) of AES, RSA and ECC
- The analysis shows that 50% of the implementations leak information
- Patches have been proposed and are currently being adopted





Future Work

- Reverse engineering of cache coherency protocols of various ARM processors
- Applying the ML techniques to recover sensitive information with low sample rate and high accuracy
- Investigate various mobile platforms like smartphones, wearables, set-top boxes and IoTs with ARM processors
- Software countermeasures to prevent the leakage of apps or to develop detection techniques using ML techniques



Bibliography

[1] Number of available applications in the Google Playstore from December 2009 to September 2016. https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store
[2] iOS App Sandbox in Depth https://developer.apple.com/library/prerelease/content/documentation/Security/Conceptual/AppSandboxDesignGuide/AppSandboxInDepth/AppSandboxInDepth.html
[3] Android Security https://source.android.com/security/

Interested in meeting the PIs? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

The 3rd NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting
January 9-11, 2017
Arlington, Virginia