Mobile Device Security: The case for side channel resistance

Gary Kenworthy and Pankaj Rohatgi

Cryptography Research Inc

As the functionality of mobile devices has increased, so have the threats. These devices make attractive targets, given the sensitivity of user and corporate data they process and store, their emerging use for viewing protected content and conducting sensitive banking and payment transactions. Until recently, hardware and software based defenses for mobile platforms lagged behind those found in more mature systems.

As a result of several high profile vulnerabilities, there have been significant commercial and research efforts directed towards improving mobile device security. Tools used for protecting servers and desktops, as well as techniques initially developed for building trusted systems are migrating into the mobile space. The use of firewalls and anti-virus products have brought desktop-level unified threat management capabilities to mobile devices [1,2,3,4]. Hardware support for secure execution environments and hypervisors are being used as building blocks for protecting sensitive applications from compromise [5,6,7]. Encrypted storage and file-systems are being used to protect data at rest [8]. Mandatory access controls at the OS and middleware layers of the mobile software stack offer further protections from potentially malicious applications [9,10,11].

While these efforts are necessary to protect mobile devices from malware and software attacks, in this *position paper*, we argue that they are not sufficient. Mobile devices are not just miniature versions of desktops and servers. Their portability and usage as a payment or identity/access token also exposes them to a larger class of attacks, including physical attacks. In this position paper, we illustrate why side-channel attacks [12,13] pose such a significant threat to these devices, by demonstrating how easy it is to extract secret cryptographic keys from these devices by monitoring their EM emissions. These attacks are completely non-invasive: the device does not have to be open or modified in any way, and in some cases these key extraction attacks can be mounted from several feet away.

Side channel Vulnerabilities of Mobile Devices

Cryptography is a basic building block for achieving security. Smartphones, tablets and other mobile device rely on cryptography for securing payments, protecting communications, protecting application and user data, protecting content, user identification etc. Many of these use cases require mobile devices to perform cryptographic operations with secret keys that must be maintained securely by the device.

However, as we illustrate through three example scenarios, with modest equipment costing around $1000 and straightforward processing, an attacker can recover cryptographic keys being used by a mobile device by analyzing its RF emissions. This problem is not limited to these three mobile devices. We have analyzed over a dozen different models of smartphones and tablets from different manufacturers, across different OSes, and they all have this vulnerability.

We now present the results from three representative devices, which we will call “Device A”, “Device B” and “Device C”. We chose not to name these devices or their manufacturers because this is an issue that is not specific to a particular manufacturer or model — it affects all devices that do not deploy countermeasures against side-channel analysis.

Analysis Configuration

Custom applications were written that perform repeated cryptographic operations in the mobile devices. This was done to highlight the security exposure, without exposing any “live keys” or specific side-channel vulnerabilities that are present in cryptographic libraries used by the operating systems in these devices. To illustrate that this is not a cryptographic algorithm specific problem, we implemented RSA in “Device A”, ECC in “Device B” and AES was used for “Device C”.

All these devices had multiple RF communications channels such as WiFi, 3G, 4G, Bluetooth, etc, but for our analysis, we disabled these RF sources by putting the device in “airplane” mode. Enabling these sources has no impact on our results: doing so only ensured that the leakages observed are from unintended EM emissions from the CPU and not from these RF transmissions.

The hardware and software requirements used to capture and analyze EM emissions from these devices are modest. These consist of an inexpensive magnetic field probe for picking up near-field emissions, a commercially available Yagi antenna for picking up far-field EM emissions, an inexpensive second-hand ICOM 7000 receiver (approx. cost $400), and a similarly inexpensive Ettus Research USRP digitizer. We used simple custom developed software as well as the open source Octave libraries for further digital processing.

Device A

Device A is a 4G LTE smart phone from a major manufacturer with an app that performs 2048 bit RSA using the Chinese Remainder Theorem (CRT). The following simplified psuedocode shows the exponentiation algorithm used for performing each of the two RSA-CRT exponentiations:

```
foreach bit i of secret exponent d (dp or dq)
   perform “Modular Square”
   if (bit i == 1) perform “Modular Multiply”
endfor
```

An M-Field probe was placed near the rear of the device and connected to the ICOM receiver tuned to 38.35 MHz. The IF output from the receiver was connected to an amplifier which was then connected to the USRP. The USRP initially sampled at 100 MHz, then reduced the bandwidth by filtering and resampling. The resulting data stream was sampled at 200 KHz.

Figure 1 shows the amplitude of the resulting signal plotted over time. Individual operations can be identified, and a “square” (corresponding to a 0) and a “square” and “multiply” (corresponding to a 1) can be easily distinguished. The secret exponents dp and dq are revealed from a a single RSA-CRT operation!

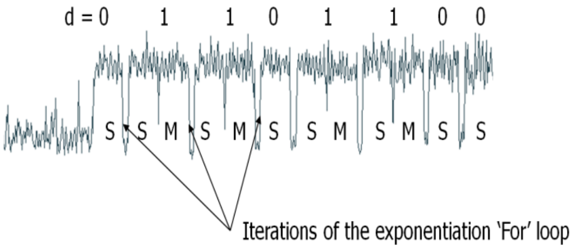

Figure 1. EM trace from Device A showing clear square and multiply operations.

Device B

Device B is a mobile PDA from a major manufacturer with an app that performs Elliptic curve point multiplication (m * Q) over P-571, for a secret scalar m, using an open source crypto library. The simplified pseudo-code for this algorithm is given below:

```
foreach bit i of secret m
   perform “Point Double”
   if (bit i == 1) perform “Point Add”
endfor
```

A Yagi antenna was placed 10 feet from the device and connected to the ICOM receiver tuned to 972.05Mhz. The IF output of the receiver was connected via an amplifier to the USRP. The USRP initially sampled at 100 MHz, then reduced the bandwidth by filtering and resampling. The resulting data stream was sampled at 200 KHz.

Figure 2 shows the resulting signal plotted over time. The individual “Point Double” and “Point Add” operations are clearly visible since they take different amount of time. Again, the secret m can be read from a single ECC operation – this time from 10 feet away.

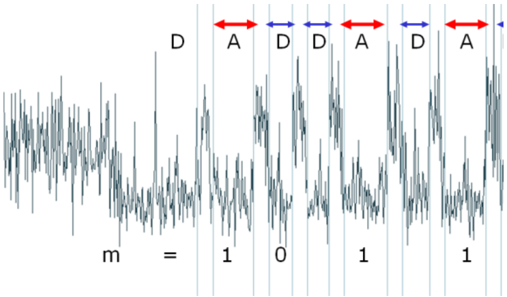

Figure 2. EM trace from Device B showing clear Double and Add operations.

Device C

Device C is another mobile phone from a major mobile manufacturer and has an app that invokes the default AES-128 library in the platform to perform AES-CBC encryption of a 200KB buffer. In this example, the M-field probe was placed behind the phone and its output was amplified and directly sampled using 60Mhz bandwidth. Figure 3 shows the signal that was obtained after some simple signal processing. The trace on the left shows the EM trace at the start of the bulk encryption, showing a repeating structure. This repeating structure corresponds to the 12500 individual AES block operations that are needed to encrypt a 200KB block of data. The trace on the right shows a zoomed-in view of the start of the bulk encryption, showing the first 3 AES operations in greater detail.

While individual AES operation may not provide enough information to reveal the secret encryption key, statistical attacks such as Differential Power/EM Analysis can aggregate information over multiple operations to amplify even the smallest leakage in noisy environments to extract keys. Figure 4 shows the results of a t-test based side channel testing technique [14], applied to EM traces corresponding to 12500 individual AES operations. Figure 4 plots the level of leakage found at different points in time of the AES operation where the X axis represents time and the Y axis represents the statistical significance of the test at point in time, measured in standard deviations over a null hypothesis that there is no leakage. Any peak outside +/- 4.5 denotes 99.999% confidence that the data set of 12500 AES operations show leakage of secret information at that point in time. Several large peaks in Figure 4, some as high as 40 standard deviations, confirm that this to be a very leaky implementation that is vulnerable to Differential EM attacks across multiple points in time during the AES operation. In fact, the larger leaks are exploitable with fewer than 1000 traces (or encryption of less than 16K of data).

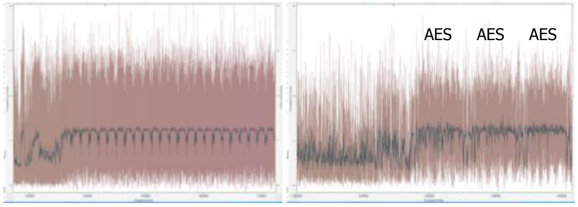

Figure 3. EM trace from Device C performing a sequence of AES block operations during bulk encryption (left) and a zoomed in view showing the first 3 AES operations (right)

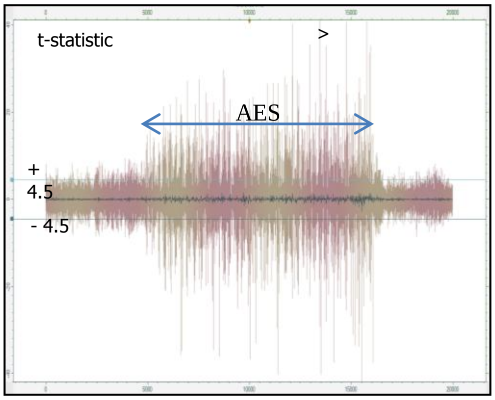

Figure 4. Result of t-statistic test of leakage using the 12500 AES operations. Several peaks exceeding +/- 4.5 indicate high degree to susceptibility.

Discussion

We have shown that any cryptographic processing being performed by the application processor of a mobile device is vulnerable to EM attacks, from distances ranges from a few cm to several feet. This vulnerability is not specific to a particular mobile device, model, or cryptographic implementation – all combinations of device/crypto library we have investigated were found to be vulnerable, unless countermeasures were present. At a minimum, this implies that content protection techniques where a user may attack the device would be vulnerable to such attacks. More worrisome are attacks that can be mounted by others: e.g., a mobile device that is using NFC for payments or physical access could be attacked by someone who places a small coil of wire near the NFC reader. In this case the crypto is performed as the device is being brought closer to the reader. Far-field attacks are even more problematic as these can be done from a distance with minimal chance of detection.

Defending against such non-invasive attacks should be part any comprehensive effort to improve the security of mobile platforms. Fortunately, this is a well-understood threat and the countermeasures that are well known and widely deployed in the financial and smart-card industries can also protect mobile devices from these attacks [12,13]. For example, modifying the RSA or ECC implementation to remove any key dependent operation sequence would eliminate the simple, single trace attacks we illustrated. To address statistical attacks, hardware implementations of crypto can be designed to minimize the data dependent EM variations. Amplitude and temporal noise can be added to reduce information within each EM trace. Techniques such as blinding and masking can randomize the data being processed by the implementation and substantially degrade the effectiveness of statistical side channel attacks. Finally, the usage of leakage tolerant cryptographic protocols for common operations such as bulk encryption or authentication can provide protection against these attacks even if the basic primitives leak.

References

1. Kaspersky Mobile Security, http://usa.kaspersky.com/products-services/home-computer-security/mobile-security
2. F-secure Mobile Security, http://www.f-secure.com/en/web/home_global/protection/mobile-security/overview
3. McAfee Mobile Security, https://www.mcafeemobilesecurity.com
4. Symantec Endpoint Protection Mobile Edition, http://www.symantec.com/sep_mobile_edition
5. ARM TrustZone, http://www.arm.com/products/processors/technologies/trustzone.php
6. Open Kernel Labs, Mobile Virtualization, http://www.ok-labs.com/solutions/what-is-mobile-phone-virtualization
7. VMWare Mobile Virtualization Platform, http://www.vmware.com/products/mobile/overview.html
8. Notes on the implementation of encryption in Android 3.0, http://source.android.com/tech/encryption/android_crypto_implementation.html
9. Sven Bugiel, Lucas Davi, Alexandra Dmitrienko, Thomas Fischer, Ahmad-Reza Sadeghi, Bhargava Shastry, Towards Taming Privilege-Escalation Attacks on Android, NDSS 2012.
10. Sven Bugiel, Lucas Davi, Alexandra Dmitrienko, Stephan Heuser, Ahmad-Reza Sadeghi, Bhargava Shastr, Practical and Lightweight Domain Isolation on Android, Proceedings of tbhe 1st ACM CCS Workshop on Security and Privacy in Mobile Devices (SPSM), 2011.
11. SEAndroid pbroject, http://selinuxproject.org/page/SEAndroid
12. Paul Kocher, Joshua Jaffe, Benjamin Jun, "Differential Power Analysis," Advances in Cryptology - Crypto 99 Proceedings, Lecture Notes In Computer Science Vol. 1666, M. Wiener, (Ed.), Springer-Verlag, 1999, pp. 388–397. (Whitepaper available at http://www.cryptography.com/resources/whitepapers/DPATechInfo.pdf)
13. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.
14. Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side-channel resistance validation, Non-Invasive Testing Workshop, NIAT 2011, http://csrc.nist.gov/news_events/non-invasive-attack-testing-workshop/papers/08_Goodwill.pdf