Defending against side-channel attacks - Part 2

Gilbert Goodwill, Cryptography Research, Inc. - March 11, 2013

Editor's Note: *This article was originally presented at ESC Boston 2011 and first appeared online on EE Times' Military Aerospace Designline.*

Part One *of this three-part series discussed a basic DPA attack against AES. Many different variations of DPA attacks exist, depending upon the type of algorithm and how it is being implemented. Part Two discusses a DPA attack against AES using EM emissions from the devices.*

---

3. EM analysis of mobile devices

3.1 Background

As smart mobile devices become ubiquitous, many applications requiring a high degree of security are being ported to the devices. Banking, mobile payments, stock trading, and digital rights management of downloaded content are all examples of applications requiring secure connections and the use of cryptographic keys. As the examples in this section show, however, many mobile devices currently in use do not contain side-channel protections. Hence, these devices are often extremely vulnerable to side-channel attacks, often from data collection occurring several yards away.

3.2 EM collection setup

The equipment used to collect the EM data is shown in ***Figure 9*** below.

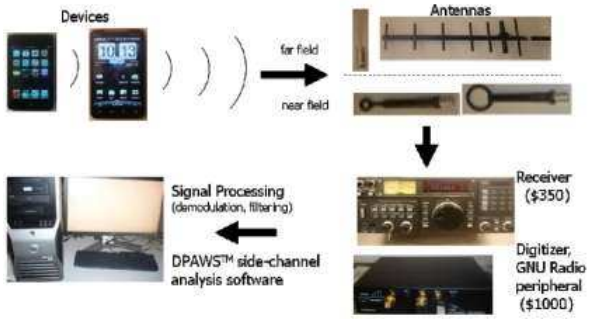

Figure 9: Setup for collecting and processing EM emissions from mobile devices

The emissions from the devices are captured with standard near field or far field antennas. The signals are sent to an Icom receiver where they are downconverted. The downconverted signals are then sent to the GNU digitizer, and the digitized signals are then processed on a standard workstation with the GNU software radio program and DPAWS software. The hardware for the entire setup can be purchased for under $2000.

3.3 EM analysis of an elliptic curve application on an iPod

The first example is an elliptic curve application running on an iPod. The application was written using an open source cryptographic library. It performs a point multiplication over the NIST curve P-521. The application computed a straightforward point multiplication using the algorithm shown in ***Figure 10*** below.

```
For each bit i of secret m
  perform "Double"
  if (bit i == 1)
    perform "Add"
  endif
endfor
```

Figure 10: Straightforward implementation of an elliptic curve point multiplication

The emissions from the iPod were collected from several feet away using the far field antenna. The carrier frequency of the signal was 972.177 MHz. The acquisition bandwidth was 200 KHz, and the filtered bandwidth was 140 KHz. A snapshot of the collected data is shown in ***Figure 11*** below.

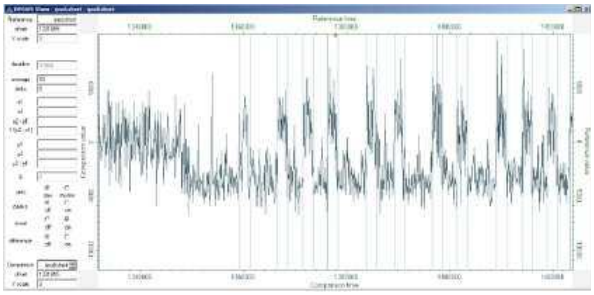

Figure 11: Data collected from iPod touch from several feet away

The double and add are very different operations, and the difference is easy to see directly. The thinner downward spikes are the doubling operations, while the wider downward spikes are the additions. In the straightforward implementation shown in ***Figure 10***, a double can be followed by either another double, or an addition. In contrast, an addition is always followed by a double. Hence, whenever there are two thin spikes in a row the corresponding bit of the secret multiplier is a zero. Similarly, a thin spike followed by a wide spike indicates the corresponding bit of the secret multiplier is a one. By analyzing the pattern of spikes, an attacker could extract the entire secret multiplier using a single trace. **3.4 EM analysis of an RSA application on an Android phone**
The second example is an RSA application running on an HTC Evo 4G phone. The application was written using an open source cryptographic library. It performs a modular exponentiation with a 2048-bit modulus, using the Chinese Remainder Theorem. The application computes a straightforward modular exponentiation using the algorithm shown in ***Figure 1***.

The emissions were collected by placing a magnetic field pickup coil behind phone. The carrier frequency of the signal was 39.99 MHz. The acquisition bandwidth was 500 KHz, and the filtered bandwidth was 250 KHz. A snapshot of the collected data is shown in ***Figure 12*** below.

Figure 12: Data collected from HTC Evo 4G phone using near field antenna placed behind phone

In these traces, the widths and heights of the multiplication and addition operations are the same. In some of the cases, however, there is a very short gap between the operations, while in other cases there is larger gap. In the straightforward implementation shown in ***Figure 1***, a square can be followed by either another square, or a multiply. In contrast, a multiply is always followed by a square. In the above trace, there are never two short gaps in a row. Hence, whenever there are two large gaps in a row the corresponding bit of the secret exponent is a zero. Similarly, a short gap indicates the corresponding bit of the secret exponent is a one. By analyzing the pattern of gaps, an attacker could extract the entire secret exponent using a single trace.

3.5 EM analysis of an AES application on an Android phone
The final example is an AES application running on an HTC phone. The application invokes the Bouncy Castle AES provider. The application performs a bulk AES encryption using a 128-bit key.

The emissions were collected with a baseband m-field trace capture on a sampling scope. The acquisition bandwidth was 100 MHz, and the filtered bandwidth was 60 MHz. A snapshot of the collected data is shown in ***Figures 13 and 14*** below.

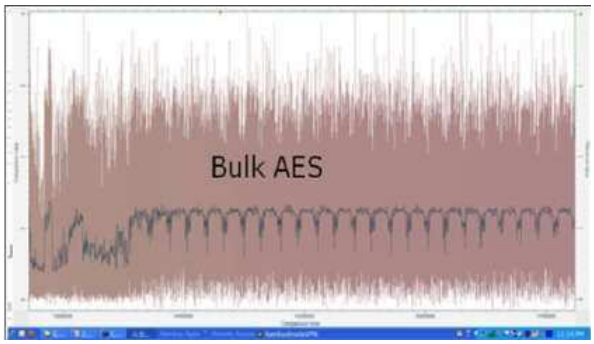

Figure 13: Data collected from HTC phone using m-field trace capture on a sampling scope

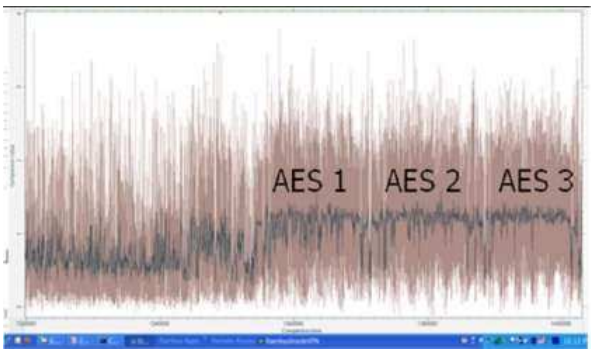

Figure 14: The individual AES operations in the bulk decryption

3.4 EM analysis of an RSA application on an Android phone
Unlike the public key algorithms in the previous two examples, the secret keys are not readily visible in the power traces. However, by applying t-tests to the traces, it is seen that the device leaks, and is likely vulnerable to side-channel attacks. (The t-tests will be discussed in detail in the next section. For now, it is sufficient to understand that the t-tests are standardized tests that indicate whether a device behaves differently based the data being processed.

In the t-test control group, the AES power traces have been randomly divided into two sets. The traces in each set are averaged, and the difference is computed. The results in ***Figure 15*** below show that there are no statistically significant differences in the two data sets.

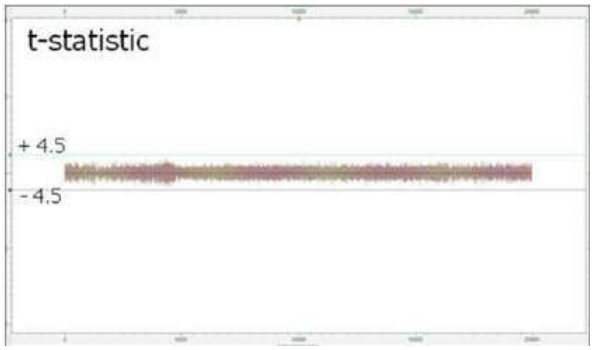

Figure 15: Control group for t-tests

In the t-test test group, one set of power traces consists of all the traces which operated on a (randomly selected) fixed 128-bit block of data. The other group consists of randomly selected traces. The results in ***Figure 16*** below show that there are statistically significant differences in the two sets of data.

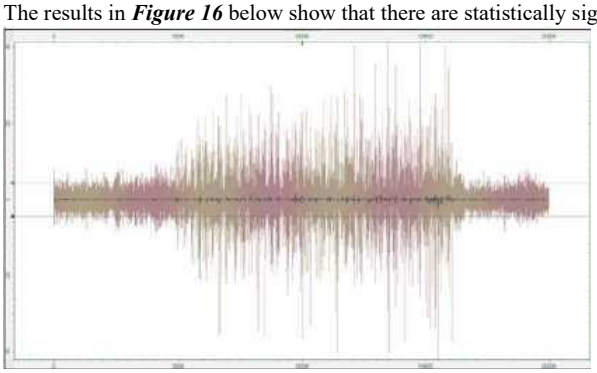

Figure 16: Test group shows device has leaks

This indicates that the device behaves differently based on the data being processed, and is likely vulnerable to side-channel analysis.

To access Part One, click here.

See related links:
Using MISRA C and C++ for security and reliability. Part I
Using MISRA C and C++ for security and reliability. Part II
Using MISRA C and C++ for security and reliability. Part III
How secure is AES against brute force attacks?
Public key cryptography and security certificates

If you found this article to be of interest, visit Military/Aerospace Designline where you will find the latest and greatest design, technology, product, and news articles with regard to all aspects of military, defense and aerospace. And, to register to our weekly newsletter, click here.