TESTBED FOR POWER ANALYSIS ATTACK BASED ON THE ARDUINO PROTOTYPING BOARD

Hasindu Gamaarachchi*, Harsha Ganegoda and Roshan Ragel

Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Sri Lanka
**hasindu2008@gmail.com*

Introduction

Embedded security devices which are used in numerous applications are quite famous in the modern world. Security of smartcards are achieved using encryption algorithms. Using the attack called power analysis even a most secure algorithm such as AES (Advanced Encryptions Standard) can be broken in few hours.

The basic setup for doing power analysis attack is shown in Figure 1. The computer sends plain text to the cryptographic device under attack and while the oscilloscope measures the power variation and send to the computer. Power measurement through an oscilloscope is realized by connecting a resistor in series to the device along the power line. After taking several thousand of readings (power traces) for different plain text samples those are analysed on a computer to derive the secret key of the cryptosystem.

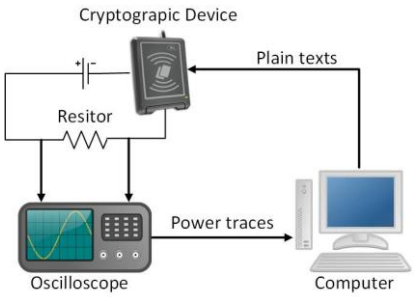

Figure 1 : Basic setup for power analysis attack

Various analysis techniques are available but the method called CPA (Correlation Power Analysis) [1] has couple of advantages such as necessity of few power traces. An intermediate value calculated during the encryption is used for the attack where for AES it is usually the S-Box lookup [2]. Hypothetical power consumption of the device is calculated using a mathematical model such as hamming weight model and then compared with power traces using Pearson correlation. The correct key is recovered using the highest correlation value. To reduce the time complexity of the attack, attack on the key happens byte by byte.

A proposed countermeasure must be thoroughly tested several times to ensure its effectiveness and the presence of any weakness which requires a setup that really does a power analysis called as a testbed. Building such a testbed is a complicated and time consuming process due reasons like the lack of a systematic guide, requirement of knowledge from multiple fields, selection of appropriate components from large number of available material in the market and debugging any issue occurred during the process. Therefore in order to address those issues we present the systematic steps for anyone who is interested in power analysis based research to build their own testbed with minimal effort and time.

Materials and methods

An *Arduino* board such as *Arduino Uno* or *Arduino Mega* where the microcontroller is removable is required while a board like *Arduino Nano* is not preferred as the microcontroller is permanently soldered. We selected an *Arduino Uno* board having an *Atmel ATMEGA328P-*

PU. Since AES (Advanced Encryption Standard) is the most used block cipher we have programmed the microcontroller to run AES with functionalities such that the microcontroller receives plain text from a computer using the serial interface and does the encryption and send the cipher text back. In order to make the power capturing process easier, a pin on the microcontroller is programmed to act as a trigger for the oscilloscope.

After programming, the microcontroller must be removed from the *Arduino* and fixed on a breadboard as shown in Figure 2. A 16 MHz crystal with appropriate capacitance provide the clock to the chip. Since the microcontroller does not have a USB controller a USB to RS232 TTL (Transistor Transistor Logic) converter has to be used. After providing power to the microcontroller and then connecting to a computer via USB the functionality can be verifying using serial port communication tools such as *TeraTerm*.

The power measurement circuit is simply a resistor of value around 100 ohms connected serially on to the power line as shown in the Figure 2. A digital oscilloscope is required where we use a *Tektronix MSO2012B* oscilloscope. An oscilloscope probe is connected to measure the variation of voltage across the microcontroller as shown there. Another oscilloscope probe has been connected to pin 4 as shown in the Figure 2 to act as the trigger.

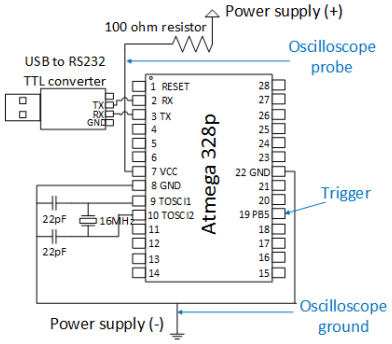

Figure 2 : Circuit diagram of the setup

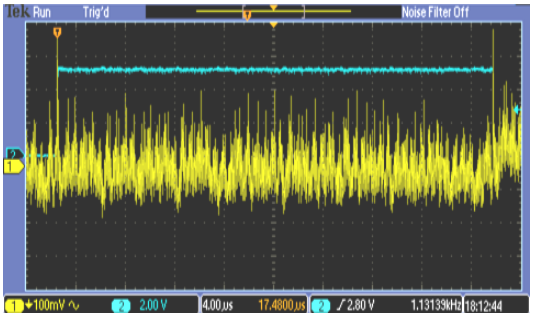

Figure 3 : Screenshot of the oscilloscope during power measurement

The power measurement process involves the computer sending several thousand of plain text and taking the readings from the oscilloscope while the microcontroller is doing encryption for each of the sent plain text. A digital oscilloscope can be connected to a computer using an interface such as USB and after installing drivers, a software such a Matlab can be used to automate the actions of the oscilloscope. We have written a Matlab script that automates the sending of plaintext and receiving of power traces.

After taking the measurements, final step is to do the statistical analysis based on the CPA algorithm explained before. Already implemented CPA algorithms are available but for improved performance we use an implementation that runs on a NVIDIA GPU. We do the analysis on a *NVIDIA Tesla C2075* GPU.

Results and Discussion

A test was done by collecting 5000 power traces. Power measurements took around 2.5 hours and the analysis took around 3 minutes. Table 1 shows results for 8 bytes of the key. The first row of the table matches the first 8 bytes of the key we used which is $0102030405060708090A0B0C0D0E0F10_{16}$. Second row shows the correlation coefficient each for those. The next rows show the keys with second and third highest correlation with values they got for correlation coefficient but those correlation values are much lesser than the maximum correlation values in row and hence the first row is clearly the correct key.

The number of power traces required is an important measure that dictates the effectiveness of a countermeasure where a plot of number of power traces vs the correlation is

useful for such purpose. Figure 4 shows such a graph drawn for a key byte. The x axis denote the number of power traces while the y axis denote the correlation coefficients. There are 256 graphs as a byte has 256 possible keys but one plot lies significantly higher than the others which is corresponding to the correct key. Even at 2000 power traces this lies significantly than the other and therefore for recovering the considered key byte even 2000 traces are enough.

Table 1 : Result for an attack on 128 bit AES using our testbed

Key 1	1	2	3	4	5	6	7	8
Correlation 1	0.146	0.140	0.168	0.126	0.091	0.112	0.144	0.140
Key 2	22	1F	5B	D5	DE	17	F2	A1
Correlation 2	0.060	0.059	0.067	0.066	0.063	0.061	0.065	0.060
Key 3	F4	89	C6	CD	A3	9	CD	1
Correlation 3	0.059	0.059	0.065	0.060	0.058	0.055	0.056	0.057

Identification of the places in the algorithms that leaks most amount of data about the key is also important in forming counter measures. Such points can be identified by plotting how the correlation coefficient varies with time. Figure 6 shows such a graphs where there x axis denote the time with respect to the number of samples in the power traces while the y axis denote the correlation coefficient. This graphs is for a key guess which turns out to be the correct key. As you can see there are sudden peak where these is the place where the writing and reading of the considered intermediate value occur.

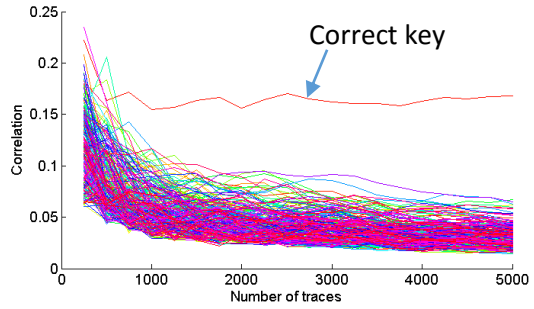

Figure 4 : Plot of correlation coefficient vs number of power traces

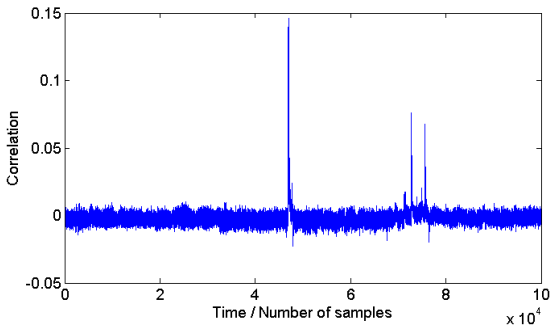

Figure 5 : Plot of correlation coefficient vs time

Conclusion

Building a testbed for power analysis is a complicated process that requires lot of effort. But the steps we present that use familiar and easy to use *Arduino* boards would make the process simple. On our testbed 128 bit AES can be broken in a less than 3 hours. The circuit being implemented on a breadboard while the *Arduino* being used as the programmer this testbed can be used easily for power analysis research such as implementing and testing countermeasures.

References

[1] E. Brier et al., "Correlation power analysis with a leakage model," in Cryptographic Hardware and Embedded Systems-CHES 2004. Springer, 2004, pp. 16–29.
[2] S. Mangard et al., Power Analysis Attacks: Revealing the Secrets of Smart Cards (Advances in Information Security). Secaucus, NJ, USA: Springer-Verlag New York, Inc.,2007.