5-2016

An Electromagnetic Coupling Model for Side-Channel Analysis

Michael L. Schena
Utah State University

AN ELECTROMAGNETIC COUPLING MODEL FOR SIDE-CHANNEL ANALYSIS

by

Michael L. Schena

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTERS OF SCIENCE

in

Computer Engineering

Approved:

Ryan Gerdes, Ph.D. Major Professor	Jacob Gunther, Ph.D. Committee Member
Koushik Chakraborty, Ph.D. Committee Member	Mark R. McLellan, Ph.D. Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2016

ABSTRACT

An Electromagnetic Coupling Model for Side-Channel Analysis

by

Michael L. Schena, Masters of Science

Utah State University, 2016

Major Professor: Ryan Gerdes, Ph.D.
Department: Electrical and Computer Engineering

This thesis presents an EM coupling model used to enhance power, side-channel measurements used in CPA. The Kalman filter is used to combine measurements of magnetic flux density with voltage or current traditionally used to measure power consumption. The DES encryption algorithm is used to evaluate CPA using EM coupled power measurements compared to traditional power measurements.

(56 pages)

PUBLIC ABSTRACT

An Electromagnetic Coupling Model for Side-Channel Analysis

Michael L. Schena

Data encryption is important for secure digital communications. While the algorithms used to encrypt data are fairly secure, the physical devices used to run them can be vulnerable. Sensitive information that can be used to decrypt data can be discovered through analyzing some of the physical attributes of these devices. Things like the power consumed by a device or the time it takes a device to perform encryption can be used to obtain this sensitive information. This thesis presents a way to us electromagnetic waves produced by a device to enhance one's ability to use the power consumed by a device to gain sensitive information.

To my family
because regardless of success or failure we all come into this world standing on the shoulders of others

ACKNOWLEDGMENTS

Firstly, I must acknowledge Dr. Todd Moon who opened the door for my first research opportunity, without which I undoubtedly would not be on the path I am today. I would like to acknowledge my advisor Dr. Ryan Gerdes for all the time and effort he invested in me throughout my development. Through the courses I took from him and his thoughtful mentoring, he truly inspired me to not only understand what things do but also discover how and why they work. Finally, I would like to thank my committee members, Dr. Jake Gunther and Dr. Koushik Chakraborty, for the time they spend for me on this thesis.

Michael L. Schena

CONTENTS

LIST OF FIGURES

ACRONYMS

CPA	Correlation Power Analysis
DES	Data Encryption Standard
AES	Advanced Encryption Standard
EM	Electromagnetic
NIST	National Institute of Standards and Technology
IP	Initial Permutation (with regards to DES)
FP	Final Permutation (With regards to DES)
FIPS PUB	Federal Information Processing Standards Publication
XOR	Exclusive Or
PC-1	Permuted Choice 1
PC-2	Permuted Choice 2
SBox	Switch Box
RTS	Rauch-Tung-Striebel
DC	Direct Current
GPIO	General Purpose Input Output
FPGA	Field Programmable Gate Array

CHAPTER 1

Introduction

The research presented in this thesis aimed to develop an electromagnetic coupling model to improve the accuracy of power consumption measurements for side channel analysis. The intent of this coupling model is to improve power measurements through noise reduction. The Kalman filter is used for integrating EM signals into power measurements. It was chosen for its capability of handling multiple signals and its design for removing the zero mean Gaussian noise that is common in physical systems [1].

Experiments were performed to evaluate EM coupling as a noise reduction technique for side channel power analysis. The main noise reduction technique used in side channel analysis is averaging multiple traces of the same operation. To analyses the noise reduction of EM coupling, CPA will be performed on traces obtained from a crypto-device with and without EM coupling. In addition, both of these implementations will utilize trace averaging. The number of traces used for averaging will be varied to compare the contributions of EM coupling to noise reduction.

DES was the first encryption algorithm to be approved and accepted by the United States government and is still used in many applications today [2,3]. It is considered to be an extremely secure encryption algorithm [4]. Although from a theoretical standpoint DES is cryptographically secure, It physical implementations have been shown to leak information through side channels and to be vulnerable to attack [5, 6]. For these reasons DES was chosen as the targeted encryption algorithm.

Two implementations of DES were implemented on embedded style devices, one microcontroller and one FPGA, and were analyzed using Correlation Power Analysis (CPA). Some preliminary work was done to assess the devices vulnerability to power analysis and then CPA was performed on measurements taken from the devices. Results of the vulnerability of these DES implementations and the effects of EM coupling will be presented.

CHAPTER 2

Background

This chapter covers the background necessary for the development of the EM coupled power model for CPA. The targeted encryption algorithm, DES, used to evaluate the CPA implementation is described. Then, side-channel analysis, power analysis, and CPA are described, giving the relationship the power consumption side-channel and DES. Finally, a general explanation of the Kalman Filter is presented; in chapter 3 the specifics of how the Kalman Filter is used to integrate EM emanation and current measurements to obtain a noise reduced power consumption measurement will be presented.

2.1 Data Encryption Standard

DES is an interoperability standard that describes the functions and formats used to communicate DES encrypted data from one computer to another [3]. DES is symmetric encryption algorithm that utilizes a shared, secret key. The algorithm was approved by NIST as FIPS PUB 46 in 1977 and was reaffirmed every five years as the standard the last of which was in 1993 [7]. DES is still in use in many applications and has been used to examine other side-channel analysis techniques (put refs here stupid!). It was the choice for this research because it is so well known and has been documented to be vulnerable to CPA.

2.1.1 Description

DES operates on a 64-bit input referred to as the plaintext and produces a 64-bit output referred to as the cyphertext. First the plaintext is passed through the Initial Permutation (IP) then split into 32-bit halves L_0 and R_0. These halves are passed through 16 rounds where R_n is passed through the Feistel Function and combined with L_n by an XOR function. At the end of each round L_{n+1} is set to R_n and R_{n+1} is set to the output of

the XOR function. After the 16th round the halves are swapped then passed through the Final Permutation (FP) which is an actually the inverse of the IP.

Each of the 16 rounds utilize a different subkey for the Feistel Function. These 48-bit subkeys are generated from the key using 2 permutation functions, PC-1 and PC-2, along with a circular shift. The key is first passed through PC-1 where it is reduced from 64-bits to 56-bits and permuted according to a set standard function. From there to produce the subkey for the nth round each 28-bit half is circular shifted left by n-bits and passed through PC-2, another set permutation function. The output of PC-2 is then the 48-bit subkey for the nth round. It is important that if any subkey is known the majority of the key can be recovered.

Feistel Function

In each of the 16 rounds, the Feistel function takes R_n and the nth round subkey as inputs. In the Feistel function R_n is expanded through what is called the Expansion Permutation, which expands the 32-bit half to 48-bits. These 48-bits are then combined with the subkey through an XOR function. This is the point where sensitive key information has entered into the encryption algorithm. In later sections how this information is taken advantage of will be explained. The XOR output is then divided into 8 6-bit words that are sent through what are called Switch Boxes or SBoxes. The combined output of these SBoxes is the output of the Feistel Function.

SBoxes

Each of the 8 SBoxes is a 4-by-16 array of 4-bit outputs. One entry of the array is chosen as the output by indexing the array with the 6 input bits. From the input, bits 0 and 5 determine the row while bits 1 through 4 determine the column. The SBoxes are important for CPA, described in later sections, because their output is determined by both the plaintext and the key but also a change of any one bit of input can result in multiple bits of output changing, which is favorable for producing distinct correlations between these bits and power consumption.

2.2 Side-Channel Analysis

Side-Channel Analysis is a branch of cryptography that analyses a crypto-system through physical attributes that leak information, which are referred to as side channels [5]. Some known side channels for DES are power consumption, scan chain [8], and electromagnetic emanations [9]. A high level block diagram of a crypto-system is shown in figure 2.2. Because crypto-systems make use of sensitive information to perform their operations, the sensitive information can affect the performance of the physical devices performing these operations. When sensitive information affects physical attributes of the crypto-system, these changes can be measured through side channels and then analyzed to potentially reveal sensitive information. If the analyzed crypto-system's side-channels directly or indirectly correspond to the system's sensitive information then a side-channel attack against that system is feasible.

2.2.1 Correlation Power Analysis

CPA is a side-channel attack that correlates the power consumption of a device when performing operation on data containing sensitive information. An example of CPA was described, attacks were mounted, and results were presented for both DES and its successor AES in [5]. For DES a register load operation, at the end of the first round or the beginning of the sixteenth round, can be targeted. CPA correlates the power consumption, of the crypto-device at that time, with the Hamming distance between some initial or reference value and a new or final value being stored in the register. Hamming distance is used because it is assumed that it correlates well with power consumption. The final value being stored must depend on both the desired sensitive information and some known variable value. If these criteria are met actual power measurements can be correlated with expected Hamming distances to infer the sensitive information contained in the data.

Equation 2.1 is the Pearson correlation coefficient that is used to calculate the correlation between the Hamming distance and the power consumption given as h and p respectively. The sample version of this equation used to compute the correlation for actual measurements is given in 2.2. The Hamming distance for a power trace i and guess g is

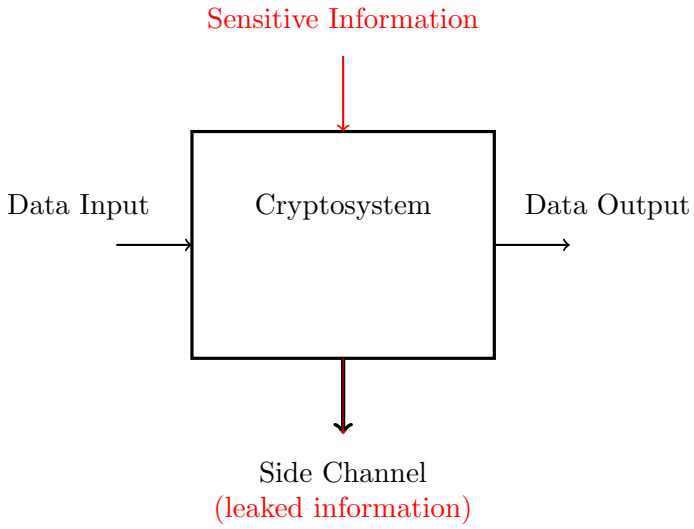

Fig. 2.1: A block diagram overview of a crypto-system that is vulnerable to side-channel analysis

given by h_i^g with $\overline{h^g}$ being the mean Hamming distance for that guess, over all n power traces. The value $p_i(t_j)$ is the power trace i at some targeted sample time t_j for which $\overline{p(t_j)}$ is the mean, again over all n power traces. The standard deviation of the Hamming distance and power consumption is s_{h^g} and $s_{p(t_j)}$ respectively.

$$\rho_{h,p} = \frac{\text{cov}(h,p)}{\sigma_h \sigma_p} \tag{2.1}$$

$$r_{h,p}^g = \frac{\sum_{i=1}^{n}(h_i^g - \overline{h^g})(p_i(t_j) - \overline{p(t_j)})}{(n-1)s_{h^g}s_{p(t_j)}} \tag{2.2}$$

The process for guessing a key using power measurements and CPA is as follows. First, exhaustively guess the sensitive bits that are to be targeted. From those guesses, for each of the known variable data used, compute the expected Hamming distance of the register operation under test. Then using 2.2 correlate the measured power with the expected Hamming distance for each guess. The guess with the highest correlation is considered the most likely guess.

Hamming Distance

Hamming distance is a measure of the difference between two binary data values. The equation for Hamming distance HD between some reference value R and a new value D is shown in 2.3. In 2.3 the XOR between R and D results in a binary number where a bit is 1 if the corresponding bit in R is different than that in D. Then the Hamming weight function HW is a count of the bits that are one in its input. This results in the Hamming distance being the count of bits in D that differ from their corresponding bit it R.

$$HD = HW(R \oplus D) \tag{2.3}$$

Hamming distance is expected to correlate with the power consumption, because the more bits are changed in a register the more power it will need to transition those bits. In the typical CMOS device, this added power consumption comes from both the cost of charging capacitances to the new static value and the dynamic cost of switching transistor states. A model for the relation between Hamming distance and power consumption in CMOS devices was developed and tested in [5]. Hamming weight and Hamming distance have been assumed or analyzed in [9–12] and has result positively.

2.3 Kalman Filter

The Kalman Filter is a widely used noise reduction filter; when the applied models and parameters are correct, it promises statistically optimal filtering [1]. This provides a useful tool for CPA because noise introduced to power measurements can greatly affect the correlation performance. Also, The Kalman Filter has been used for utilizing the information from multiple types of measurements to approximate a single desired parameter or to produce some other aggregate parameter [13–15]. This is why it was chosen for the practical purposes of this work that entails combining magnetic flux measurements with supply current measurements to better approximate a device's power consumption. The rest of this section will layout general Kalman filter method that was used in the implementation of experiments for this thesis.

2.3.1 Process Model

In order to use the Kalman Filter to approximate the state of some process a model of the process and the measurements of said process must be made. The equations 2.4 and 2.5 give discrete models for the process state x_k and its measurements z_k. In the Process Model the vector u represents the known control inputs to the system and those inputs are transformed into the process state space by the matrix B. The Process state vector x_k hold the observed and unobserved parameters of the process that are being modeled. The matrix F represents how the process develops from one timestep to the next. In a trivial example where the process state vector has two parameter that are assumed to be constant F would be given by 2.6, a two by two identity matrix. This example may seem unimportant, but if the parameters are relatively constant with a small enough time step it may be appropriate to model them this way. The measurement vector z_k holds the observations of system parameters and the matrix H describes how the system parameters relate to the measurement values. w_k and v_k are the process noise and measurement noise respectively. Both are assumed to be zero mean Gaussian random variables. Estimating the variance of these noise variables is part of deriving a Kalman Filter for any specific system.

$$x_k = Fx_{k-1} + Bu_k + w_k \tag{2.4}$$

$$z_k = Hx_k + v_k \tag{2.5}$$

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.6}$$

2.3.2 Filter Update Equations

From the process model the Kalman Filter produces its estimation and correction equations shown in 2.7 and 2.8 respectively. Equation 2.7 estimates the system state $\hat{x}_{k|k-1}$ with all the information known up to and including timestep $k-1$. Then, 2.8 corrects that estimate using the new information of z_k measured at timestep k. To correct the estimation 2.8 multiplies the difference between the estimated state and the measured state known as

the residual by the Kalman Gain K_k to determine how much estimate needs to be corrected based on the measurement. To filter a set of measurements the process of estimating and then correcting is repeated for each timestep.

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + Bu_k \tag{2.7}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{2.8}$$

2.3.3 Filter Parameters

In order to use the Kalman Filter the Kalman Gain must be computed for each time step. Equation 2.10 gives the update equation for K_k that depends on the Covariance Matrix P and Measurement Noise Covariance R. In equations 2.9 $Pk|k-1$ is estimated using the previous Covariance Matrix and the Process Noise Covariance. The Process Noise Covariance used here Q_k is a time varying approximation based on measurements. The details of how the noise parameters R and Q_k were determined will be given in the implementation chapter. Finally the Covariance Matrix is updated based on the Kalman Gain that was computed for this time step. The equations 2.9-2.11 shows that K_K is based on the variance of both process and measurement noise and represents a ratio of uncertainty between the state estimation and state measurement.

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k \tag{2.9}$$

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^-1 \tag{2.10}$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \tag{2.11}$$

2.3.4 Rauch-Tung-Striebel

The RTS smoother was first developed in [17]. It is a smoother based on Maximum-Likelihood that can be added to the Kalman Filter. RTS consists of two passes, a forward pass that is the original Kalman Filter where the a-posteriori and a-priori state vectors and

covariance $x_{\hat{k}|k}$, $x_{k|\hat{k}-1}$, $P_{k|k}$, and $P_{k|k-1}$ are stored for a backward pass. In the Backward pass $x_{\hat{k}|n}$ is computed as shown in 2.12. This computation uses a new Gain factor C_k determined by the a-posteriori and a-priori covariance. The RTS smoother was applied when filtering data in this research.

$$x_{\hat{k}|n} = x_{\hat{k}|k} + C_k(x_{\hat{k+1}|n} - x_{\hat{k+1}|k}) \tag{2.12}$$

$$C_k = P_{k|k}F_{k+1}^T P_{k+1|k}^{-1} \tag{2.13}$$

$$P_{k|n} = P_{k|k} + C_k(P_{k+1|n} - P_{k+1|k})C_k^T \tag{2.14}$$

CHAPTER 3

Kalman Theory

This chapter covers the details of the specific Kalman Filter used for the implementation of an EM coupled CPA attack. The main intention of this filter is to improve the power measurement by removing as much measurement noise as possible. To improve the noise reduction, the RTS smoother was used after the forward pass of the Kalman Filter is completed as described in chapter 2. There are multiple methods by which sensor measurements can be combined using a Kalman Filter [16]. In this work measurement were simply combined through the observation matrix. The process model and noise estimation used to apply the Kalman Filter to EM coupled power measurements in this work are described in this chapter.

3.1 Electromagnetic Coupling

Power analysis is a type of side channel analysis that gains leaked information from the power consumption of a device. Figure 3.1 shows an EM coupled model for measuring the power consumption of a targeted device. In this model the device is assumed act like a wire for the purposes of EM measurements. In order to couple a devices EM emanations to its power consumption the relation between current and magnetic flux is used. $\frac{\delta \mathcal{B}}{\delta t}$ is a proportional measure of $\frac{\delta I}{\delta t}$ where $\mathcal{B}$ is the magnetic flux and I is current. The infinitely long wire approximation, $\mathcal{B} = \frac{\mu_0 I}{2\pi r}$, is used. The antenna used to probe the device measures the change in magnetic flux density $\mathcal{B}$ [18].

3.2 Process Model

The process model developed here represents a crypto-device and tracks the total power of the device. The measurements represent the voltage over a shunt resistor inserted in the ground line of the supply voltage and the change of magnetic flux density caused by current

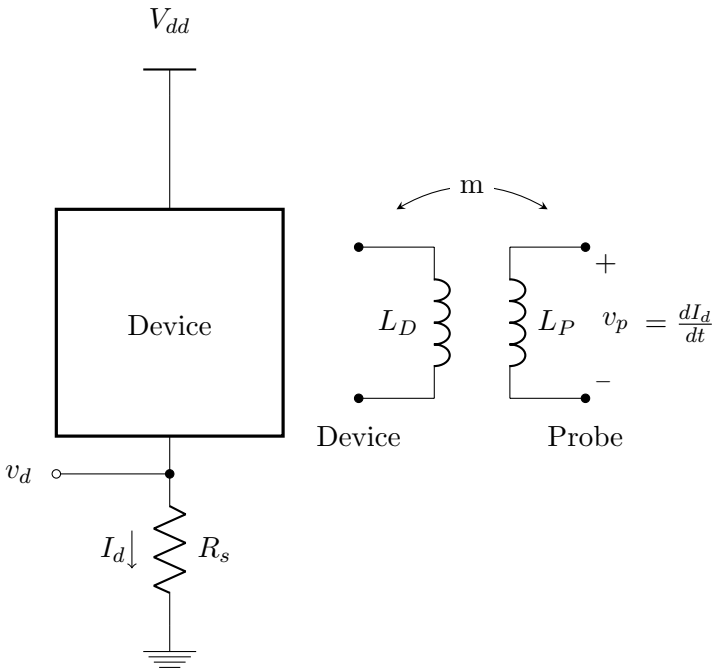

Fig. 3.1: EM Coupling model V_d measurements shown on left and EM probe model shown on right

passing through the shunt resistor. The equations 3.1 and 3.2 give models for the process state x_k and its measurements z_k. For this model it is assumed that the control vector input u is null and so it and B are excluded. The Process state is represented as $x_k = [V_k \ \dot{\mathcal{B}}_k]^T$ where V and $\dot{\mathcal{B}}$ respectively are proportional to I and $\dot{I}$ from the EM coupling model. The process model used here assumes $\dot{\mathcal{B}}$ is constant and I is given by the previous value for I plus $\dot{I}$ times the timestep Δt, which leads to process matrix given in 3.4. The measurement is $z_k = [v_k \ \dot{b}_k]^T$ where v_k is a voltage measurement that is proportional to I_k by some resistance value r and $\dot{b}_k$ is the measured change in magnetic flux which is proportional to $\dot{I}$ by some coupling coefficient m. This results in the equation for H given by 3.3.

$$x_k = Fx_{k-1} + w_k \tag{3.1}$$

$$z_k = Hx_k + v_k \tag{3.2}$$

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.3}$$

$$F = \begin{bmatrix} 1 & rm\Delta t \\ 0 & 1 \end{bmatrix} \tag{3.4}$$

3.3 Filter Execution

The equations used execute the Kalman Filter are its estimation and correction equations shown in 3.5 and 3.6 respectively. Equation 3.5 estimates the system state $\hat{x}_{k|k-1}$ to be $[V_{k-1} + \dot{B_{k-1}}\Delta t \ \dot{B_{k-1}}]^T$ where V and $\dot{\mathcal{B}}$ come from the a-posteriori state vector $x_{k-1|k-1}$. Then, 3.6 computes the residual, the difference between $x_{k|k-1}$ and z_k. To correct the estimation 3.6 multiplies the difference between the estimated state and the measured state known as the residual by the Kalman Gain K_k to determine how much estimate needs to be corrected based on the measurement. To filter a set of measurements the process of estimating and then correcting is repeated for each timestep.

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} \tag{3.5}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \tag{3.6}$$

For each timestep the Kalman Gain is computed based on the process and measurement models, specifically F and H, developed in section 3.2, and Q and R, which will be developed in section 4.3. Equation 3.8 gives the update equation for K_k that depends on the uncertainty covariance Matrix P and Measurement Noise Covariance R. In equations 3.7 $Pk|k-1$ is estimated using the previous Covariance Matrix and the Process Noise Covariance. The Process Noise Covariance used here Q_k is a time varying approximation based on measurements. The details of how the noise parameters R and Q_k were determined will be given in the implementation chapter. Finally the Covariance Matrix is updated based on the Kalman Gain that was computed for this time step. The equations 3.7-3.9 shows that K_K is based on the variance of both process and measurement noise and represents a

ratio of uncertainty between the state estimation and state measurement.

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q_k \tag{3.7}$$

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-}1 \tag{3.8}$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \tag{3.9}$$

CHAPTER 4

DES Power Data Collection

To test the proposed EM coupling model developed in Chapter 3 magnetic flux density and voltage measurements, referred to as power traces or records, of a crypto-device have to be taken. This chapter describes the experimental set up used to record measurements and some details of how the measurements were improved. To improve the quality of magnetic flux density measurements, some EM shielding was constructed and placed around the antenna probe along with the targeted component. After collecting the data some preprocessing was done to align the records and determine the samples to target for CPA.

Aside from the power traces used for CPA, measurements were taken for the estimation of process and measurement noise needed for the Kalman Filter implementation. Measurements for the process noise were taken much like the CPA power traces but the key is varied along with the plaintext. The Measurement noise was analyzed using an arbitrary function generator in place of the cryptodevice. The function generator was used in order to have a more precisely known signal to measure. All the Data collection is described further in the following sections.

4.1 Data Collection

For execution of CPA on a device power traces of the device need to be recorded while the device is performing it's encryption algorithm on variable inputs (plaintext) using the same sensitive information (key). For this experiment 1000 different plaintexts used for encryption with the same key. In addition, 125 different power traces were taken for each plaintext in order to be averaged, which is a common noise reduction technique used for CPA. To automate the process a GPIO pin was lowered just before DES began and raised again at the end. Also, a delay loop was used in between DES iterations to allow for data transfer from the oscilloscope to a computer, which was storing the data.

Collection of the power traces was done with a Techtronics MDO 4104-3 oscilloscope. Each DES trace consisted of 5000000 samples taken at 625 mega-samples per second. Three channels of the oscilloscope were used one for the power voltage, the EM antenna probe, and a probe on the GPIO pin trigger. The voltage probe was placed above a 9.856 Ohm shunt resistor on the ground line between the device and an external power supply. The EM nearfield antenna was positioned above the shunt resistor, both within a Ferriday cage, and its voltage signal was passed through a 10 Db amplifier before reaching the oscilloscope. Examples of voltage and magnetic flux traces are shown in figures 4.1 along with the traces after averaging over 125 traces in figures 4.1.

4.2 EM Shielding

Because of the nature of probing EM signals, noise is present from many external signals. It was shown in [19] that shielding an EM probe is an effective way of reducing noise from external sources. As in [19] a Faraday cage was constructed to shield the probe. The cage was a four by four by four inch cube, pictured in figure 4.2, constructed with two mesh layers, one of Steel and one of Aluminum. Apertures had to be cut to allow for the EM probe and device components to be connected while inside the Cage. The holes were kept as small as practical. When in use, The Ferriday cage was connected to the external power supply's ground line below the shunt resistor.

4.2.1 Alignment

Alignment was performed on the records to compensate for the trigger jitter caused by the oscilloscope's edge triggering device. A matched filter was used as the alignment technique where each record was aligned to a reference record from which the matched filter was produced. A distinctive oscillating in the voltage signal, caused by raising the GPIO trigger, was used as the matched filter and cross correlation was computed between the Matched filter and each record. A peak and oscillations appeared in the cross correlation output, example shown in figure 4.2.1, corresponding to the position of the voltage oscillation in each signal. As apparent in Figure 4.2.1, the two first positive peaks are fairly

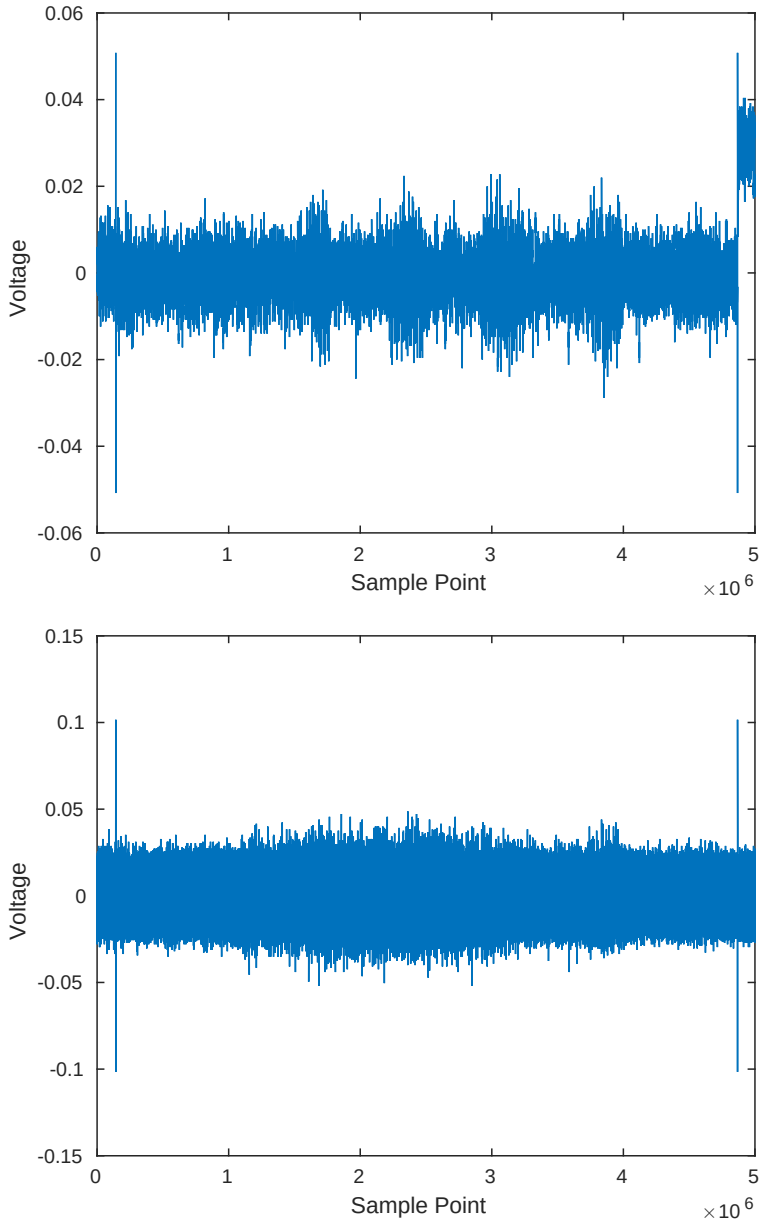

Fig. 4.1: A typical power (top) and EM emanation (bottom) trace taken while the Tiva performs DES

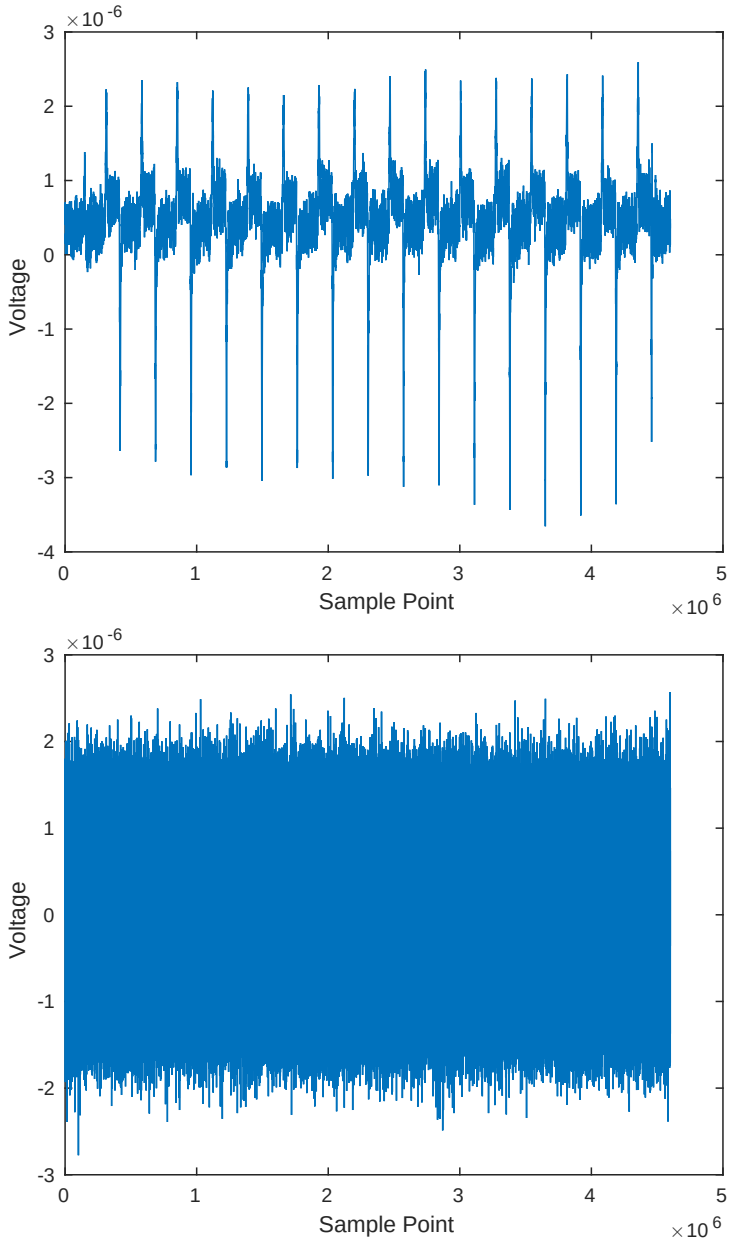

Fig. 4.2: A power(top) and EM emanation(bottom) trace resulting from averaging 125 power traces taken while encrypting the same plaintext with the same key

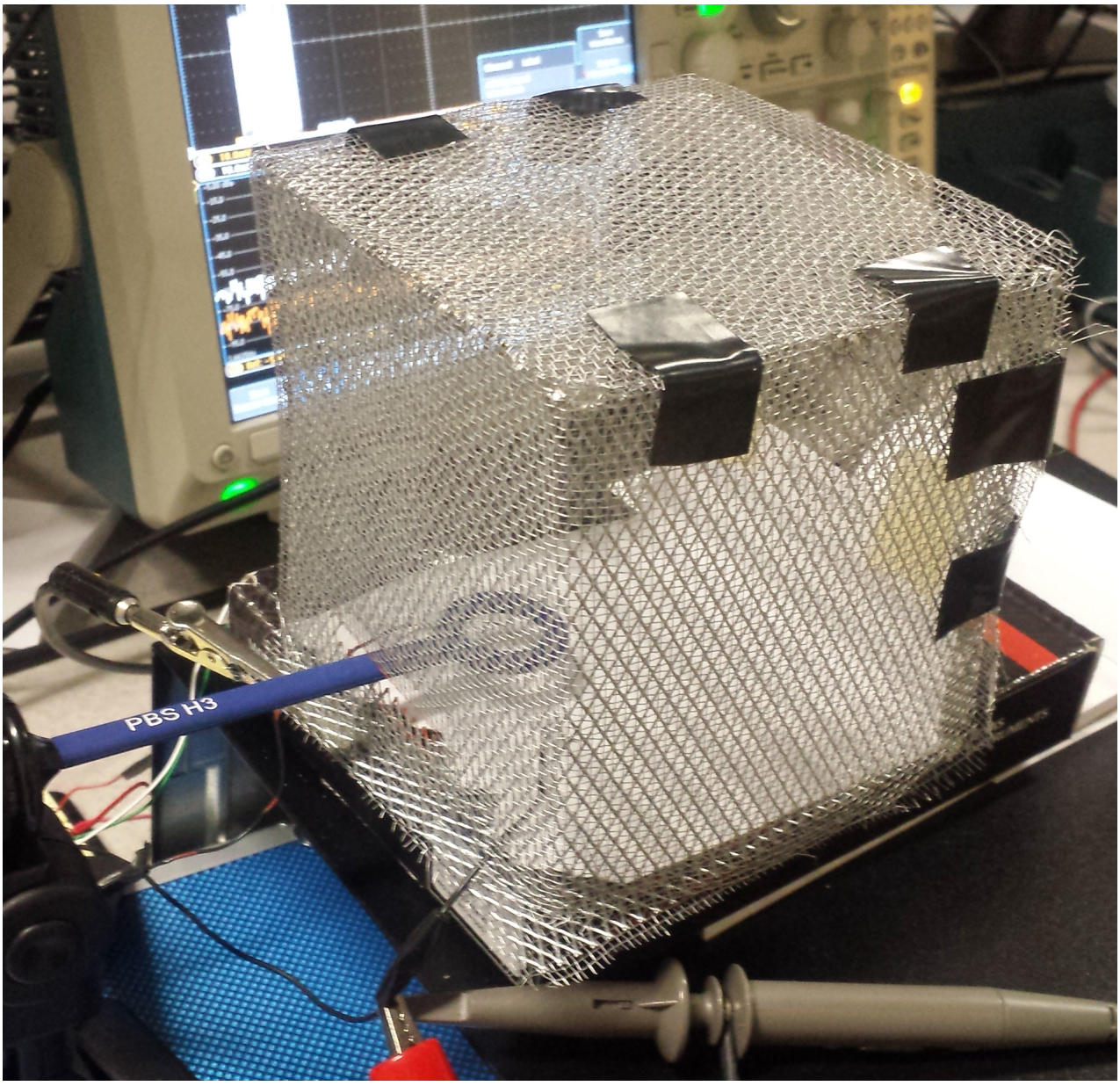

Fig. 4.3: The Faraday Cage with EM probe and Shunt Resistor Inserted

close in magnitude, so because the position is going to be used to create a relative offset the minimum point of the negative peak was used. After finding these position they were normalized by subtracting the position for the reference trace to determine the relative offset. When averaging was performed as explained in chapter 5 each trace was shifted by the offset index determined through this process.

4.3 Noise and Variance

The two sources of noise that are modeled in the Kalman process model are the process and measurement noise given respectively as w_k and v_k in 3.1 and 3.2. When executing the Kalman filter these two noise sources are expressed through Q and R that are used to determine the Kalman Gain K_k for each timestep. These two parameters may need to be estimated for each specific crypto-device, encryption algorithm, or measurement setup. This section explains how for the experimental setup used in this thesis.

4.3.1 Process Noise

In order to approximate the Process Noise, the variance is measured across the process operating on different inputs. The two inputs that affect the process are the plaintext and the key. To measure the process noise one hundred and twenty five power and magnetic flux traces were taken for each of one thousand plaintext, key pairs. Each of these sets of one hundred and twenty five traces were then averaged. Then for each timestep, the covariance between the voltage and magnetic flux of the thousand averaged traces was computed. The result is a covariance matrix Q_k for each timestep of the process. The resulting Q is shown if figure 4.3.1.

In practice the process noise estimated using this method could be determined using a device identical to the target device an attacker has access to beforehand. If fact, the process noise estimation should be applicable for any identical device running the same encryption algorithm, and so could be used for multiple attacks on separate devices. Although this method takes a substantial amount of time and traces, because it can be done beforehand and without access to the targeted device it does not increase the time of performing the

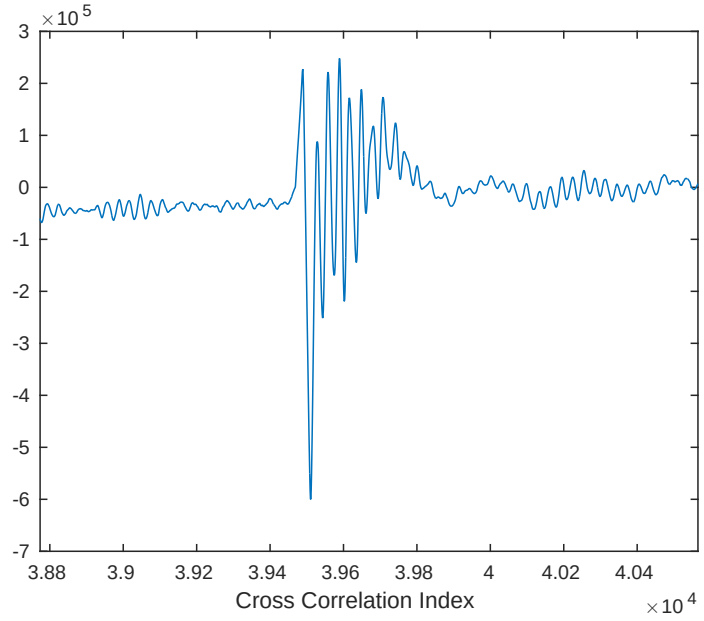

Fig. 4.4: Output of the cross-correlation between a matched filter and a record to be aligned

actual attack.

$$Q_k = \mathrm{cov}(\mathbf{v}_k, \mathbf{b}_k) \tag{4.1}$$

$$\mathbf{v}_k = [v_{1,1}(t_k)\, v_{2,2}(t_k) \cdots v_{n,n}(t_k)] \tag{4.2}$$

$$\mathbf{b}_k = [b_{1,1}(t_k)\, b_{2,2}(t_k) \cdots b_{n,n}(t_k)] \tag{4.3}$$

$$Q_k = \mathrm{cov}(\mathbf{v}_k, \mathbf{b}_k) \tag{4.4}$$

$$\mathbf{v}_k = [v_{1,\star}(t_k)\, v_{2,\star}(t_k) \cdots v_{n,\star}(t_k)] \tag{4.5}$$

$$\mathbf{b}_k = [b_{1,\star}(t_k)\, b_{2,\star}(t_k) \cdots b_{n,\star}(t_k)] \tag{4.6}$$

4.3.2 Measurement Noise

The variance of voltage and magnetic flux density measurement noise must be estimated to successfully apply the Kalman Filter. The Variance of the estimated noise is used in the R matrix used in equation 3.8; R is given by 4.7 where σ_V and $\sigma_{\dot{\mathcal{B}}}$ denotes standard

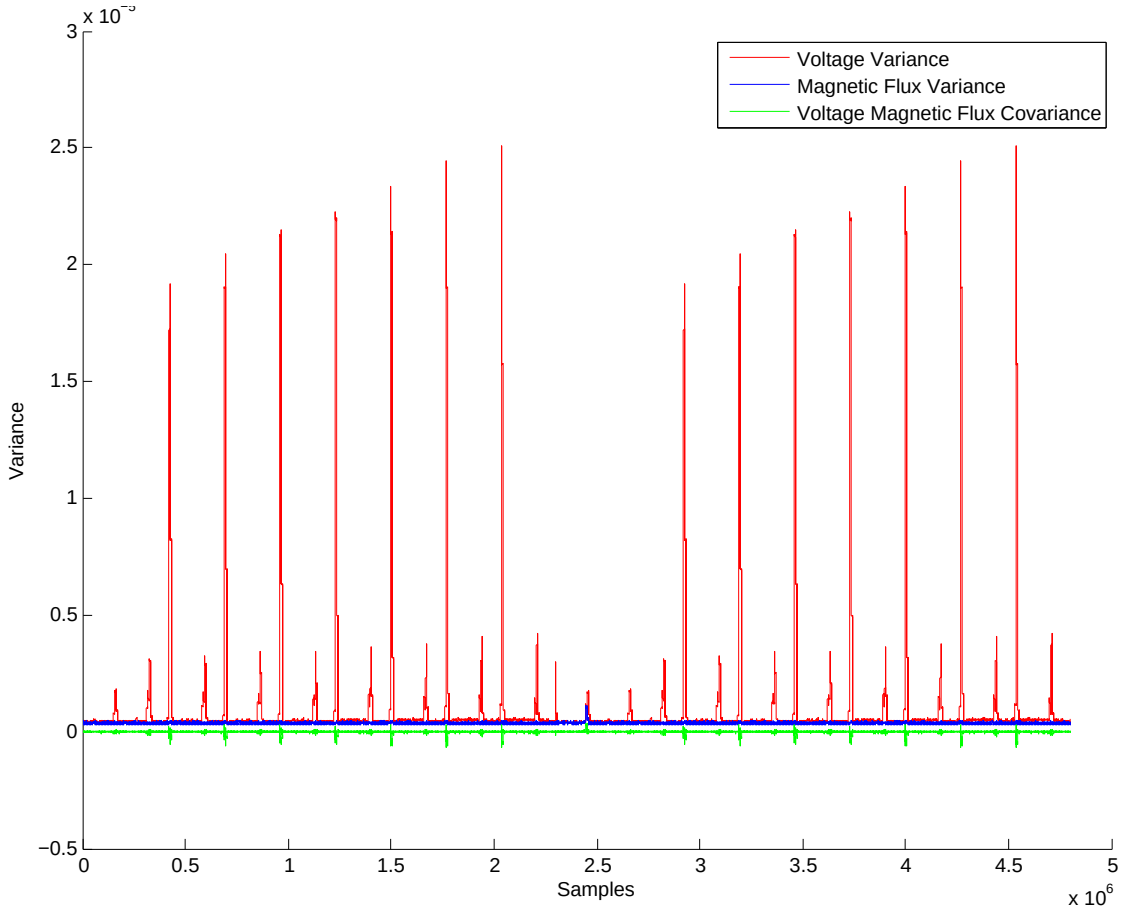

Fig. 4.5: The time varying Q used in Kalman Filter

deviation of voltage and magnetic flux density measurement noise respectively. Also, they must be determined separately because of differences in the measurement methods. The Voltage measurements were taken with an oscilloscope probe connected to the shunt resistor terminal. Differing from this, the Magnetic flux measurements were taken with an antenna positioned above the shunt resistor. Also, the Magnetic flux signal was passed through a 10 Db preamplifier before being passed to the oscilloscope.

$$R = \begin{bmatrix} \sigma_V^2 & 0 \\ 0 & \sigma_{\dot{\mathcal{B}}}^2 \end{bmatrix} \tag{4.7}$$

In addition, the measurements are being averaged before being input to the Kalman Filter and the estimated noise needs to reflect that. Because the noise is assumed to follow a zero centered Gaussian distribution averaging should reduce the variance of measurement noise present in the signal. To simulate this, while estimating the noise, multiple traces were taken, then some number of them are averaged depending on how many traces per plaintext are being averaged.

After implementing the filter with the estimated noise variances described in this section, the filter was relying too much on z_k for each V_k estimate, following within 10^{-6} volts of the measurement at each timestep. There are a few possible reasons this discrepancy. First, the physical set up was slightly different when taking the records for Noise estimation than when taking DES power traces. This was because the arbitrary function generator took place of both the crypto-device and the power supply, which reduced the number of connections between the voltage process and the oscilloscope probe. Another reason could be that the DES power traces had a DC offset whereas the Measurement Noise was estimated using a constant zero volt signal. After considering these differences, the voltage measurement standard deviation σ_V was increased to ten times the original estimate.

Coupling Coefficient

The magnetic flux density traces used to estimate the measurement noise were also used to estimate the coupling coefficient m necessary for equation 3.4. The signal being

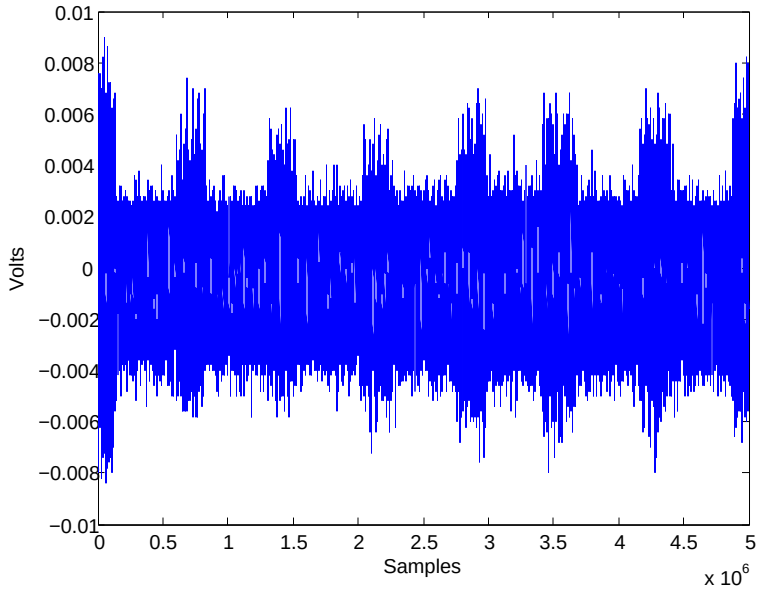

Fig. 4.6: An Example Voltage Trace taken of a constant zero volt signal produced by a function generator across the shunt resistor

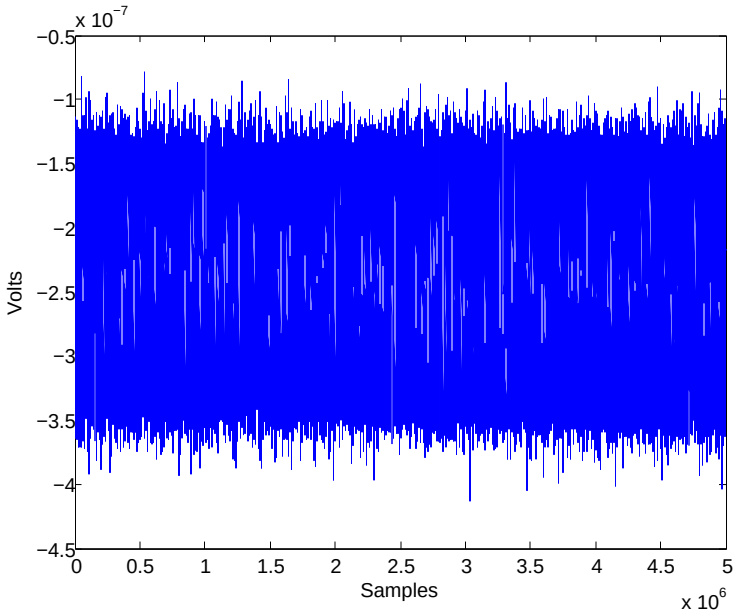

Fig. 4.7: Voltage Trace after averaging 125 traces used to compute standard deviation

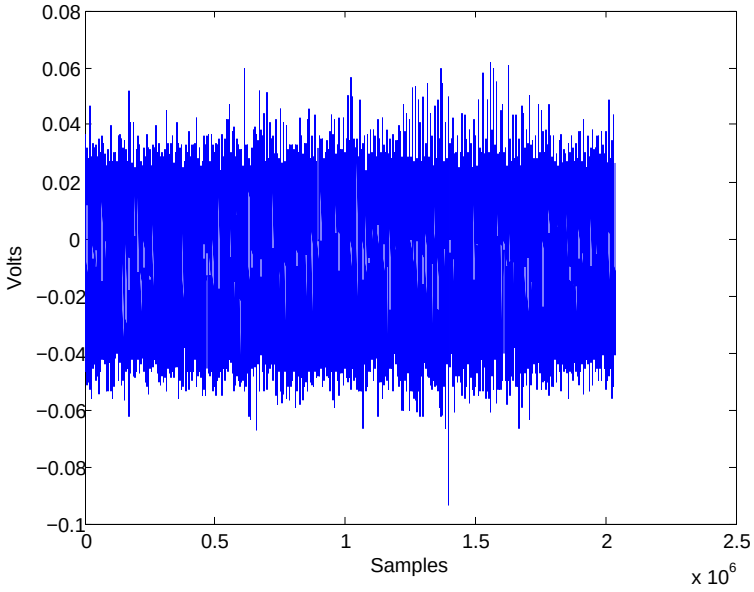

Fig. 4.8: An example EM emanation trace taken over a constant-slope signal across the shunt resistor

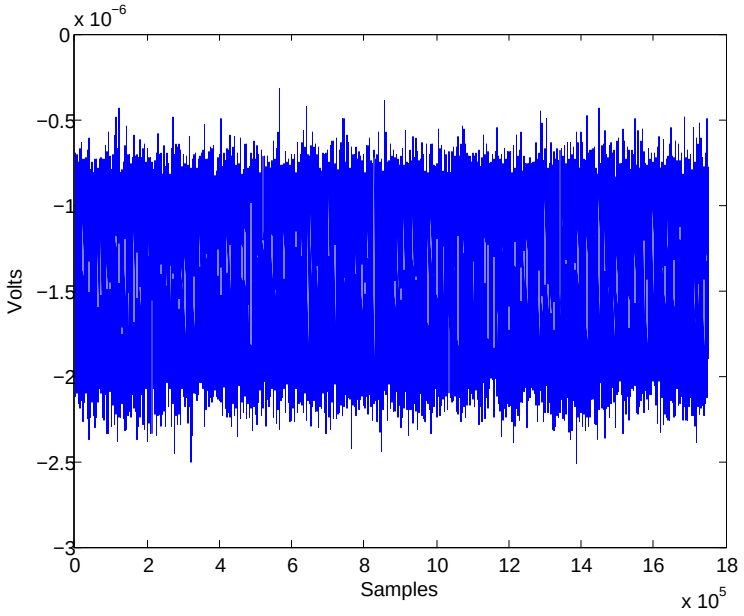

Fig. 4.9: Magnetic flux signal after averaging 125 traces used to compute standard deviation

measured was EM emanations from the shunt resistor with a constant slope current signal passing through it. This should result in a constant change to the magnetic flux density, measured as constant voltage across the antenna probing the magnetic field. The ratio between the mean of the measured signal and the current slope was computed and used as m.

CHAPTER 5

Tiva-C CPA Implementation

This Chapter covers the implementation of CPA and the Kalman Filter used to analyze the coupling of magnetic flux and voltage measurements to measure power consumption. An experiment was performed where measurements were taken of the electromagnetic emanations and supply current of a microcontroller while it was performing DES encryption. These measurements were analyzed using different implementations of CPA including one that utilizes the Kalman Filter. The Results of each CPA implementation are then presented and compared.

This chapter will cover the implementation details as follows. First, the Hamming distance power model is validated for the targeted device. Then, the Experimental setup used to take electromagnetic and voltage measurements is explained followed by the CPA implementation applied to those measurements. Finally, the results of each implementation will be presented and analyzed.

5.1 Power Model Validation

In order to know whether or not CPA is for a DES implementation on a specific device, a correlation between the power consumption and the Hamming distance of the registers of the device under test must be shown. A program was written in c and loaded on a Texas Instruments Tiva-C Series TM4C123G microcontroller to examine its register power consumption. Some difficulties are present with the Tiva-C. The Tiva-C contains a pipelined processor which means power is being consumed for multiple instructions at the same time. Also, the registers and other hardware of the Tiva-C are smaller than FPGA's which these types of attacks are commonly demonstrated on. These difficulties make the Tiva-C a challenging target for showing possible improvements to existing attacks like CPA.

Power measurements were taken to confirm the Hamming distance power model. The

voltage was probed over a 10 Ohm resistor inserted between the USB ground wire and the external power supply ground. Records of length 100000 were taken at a rate of 50 GS/s. Oscilloscope was triggered using a GPIO pin. One thousand records were taken using each Hamming distance 0 through 32. The Program running on the microcontroller was essentially the pseudo-assembly program shown in 5.1. In this program the GPIO trigger, accumulator, and branch instructions are independent of the values stored in R1 and R0, so they should have the same effect on every trace indifferent to the Hamming distance being tested. Also, examining the move instructions on line 9 and 10 using 2.3, where R is R0 before instruction execution and D is R0 afterwards, the Hamming distance for both instructions should be the same and equal to the Hamming weight of R1.

After measuring, traces were averaged over the 1000, 500, 250, and 125 records. The beginning section of the record was not used in averaging because there were oscillations due to the GPIO pin being switched. Examples of a signal trace and an averaged trace is shown in figure 5.1 and 5.1 respectively. After averaging the traces, the mean voltage value was calculated over different lengths of the trace.

Observations

The power vs Hamming distance graph shows a positive correlation between voltage and Hamming distance. When averaging over 1000 measurements and 90000 samples Voltage is strictly increasing with Hamming distance. A plot of the resulting voltage curve is shown in figure 5.1. One intriguing result of the measurements is the slope of voltage for a Hamming distance of 16 to 32 was about double that from 0 to 15. It would be interesting to look into the cause of this difference and examine if it has some exploitation. Even though power has a strictly positive correlation with Hamming distance, from the magnitude shown here and the number of instructions needed to average we predict that the baseline CPA implementation will not work, due to the resolution and number of samples taken.

```
1 R1 = #DesiredHD ;0x0, 0x1, 0x3, 0x7, ..., 0xFF were used
2 loop:
      GPIO = 0
4     bl delayloop
      GPIO = 1
6     R2 = #0x0
  innerLoop
8         add R2, #0x1
          mov R0, R1
10        mov R0, #0x0
          cmp R2, 0xFFFF
12        beq loop
          b innerLoop
```

Fig. 5.1: Psuedo assembly code of the program used to validate Hamming distance power model

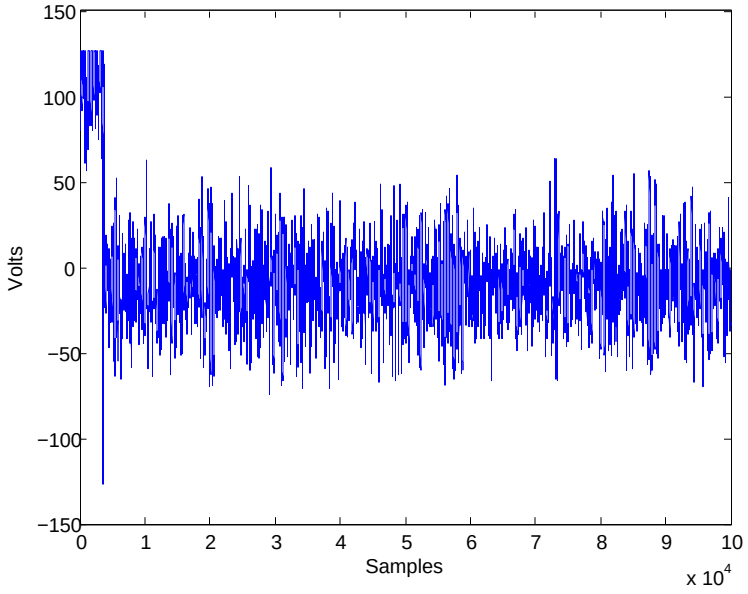

Fig. 5.2: Example trace of a power trace recorded while running the assembly program in figure 5.1 with a Hamming distance of 0

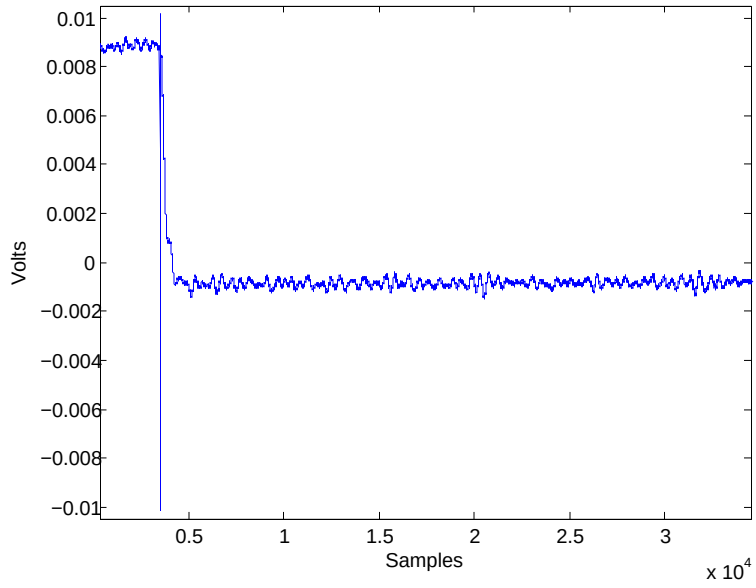

Fig. 5.3: Result of averaging 125 power traces used to test the Hamming distance power model

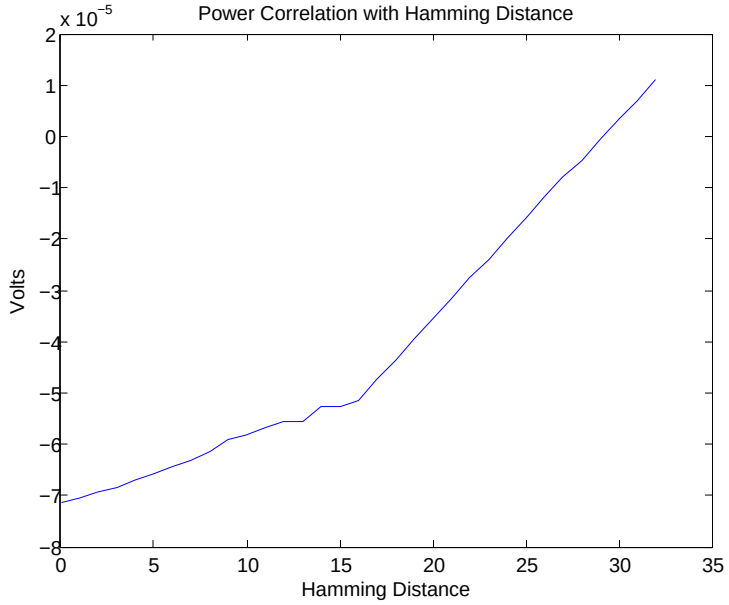

Fig. 5.4: The mean voltage of averaged traces vs the Hamming distance of register operations being performed

5.2 Correlation Power Analysis

The targeted Hamming distance must be affected by the key, but also be able to change when a different plaintext is used.

This CPA implementation targets two sets of instructions of the DES implementation that handled the SBox output. Because the SBoxes produce 4-bit outputs every two SBoxes different instructions are used to handle the upper nibble and lower nibble of the four output bytes. Both of these code segments contain instructions that are suitable for being targeted by CPA.

The section of code listed in figure 5.2 shows the assembly used handle the SBox output that creates the upper half of each byte of Feistel function output. The important instructions are listed on lines 11 through 13. Line 11 loads the SBox output into register r0. Then, lines 12 and 13 shift the data left by 28 bits and back 24 bits to put it in the correct position. The Hamming distance of r0 across both of these instructions equals two times the Hamming weight of the SBox output.

The second section of code that handles the lower half of a byte output is shown in figure 5.2. Here the instruction on line 4 loads the SBox output into r1. Then, on line 5 the 4 bites in r1 are combined with r0 using an ORR instruction. In this case the ORR instruction can be used, because the lower 4-bits of r0 is known to be all 0's before the instruction execution and equals the SBox output afterwards. This results in a Hamming distance is equal to the Hamming weight of the SBox output.

5.2.1 Correlation Model Validation

To find the location where sensitive information is present in the power trace, Correlation is performed for every sample point of the DES encryption using only the correct Hamming distance for each trace instead of every guess. From the symmetry of the 16 rounds as seen in figure 4.1, one can infer the general location of the first round. Unfortunately, no strong correlation peaks presented themselves in the expected area shown in figure 5.2.1 or elsewhere. The correlation between correct Hamming distance and the entire first half of DES execution is shown in figure 5.2.1.

```
 1 F89D0004   LDRB          r0,[sp,#0x04]
 2 F0000080   AND           r0,r0,#0x80
   1184       ASRS          r4,r0,#6
 4 F89D0004   LDRB          r0,[sp,#0x04]
   F3C00080   UBFX          r0,r0,#2,#1
 6 4304       ORRS          r4,r4,r0
   F89D0004   LDRB          r0,[sp,#0x04]
 8 F3C006C3   UBFX          r6,r0,#3,#4
   48EE       LDR           r0,[pc,#952]  ; @0x000007DC
10 EB001004   ADD           r0,r0,r4,LSL #4
   5D80       LDRB          r0,[r0,r6]
12 0700       LSLS          r0,r0,#28
   0E00       LSRS          r0,r0,#24
14 F88D0000   STRB          r0,[sp,#0x00]
```

Fig. 5.5: The instructions shown here load an SBox output into the upper four bits of each Feistel function output byte

```
 1 F89D0000   LDRB          r0,[sp,#0x00]
 2 49E1       LDR           r1,[pc,#900]  ; @0x000007E0
   EB011104   ADD           r1,r1,r4,LSL #4
 4 5D89       LDRB          r1,[r1,r6]
   4308       ORRS          r0,r0,r1
 6 F88D0000   STRB          r0,[sp,#0x00]
```

Fig. 5.6: The instructions show here load the SBox output into the lower half of the Feistel function outputs. based on a six bit SBox input

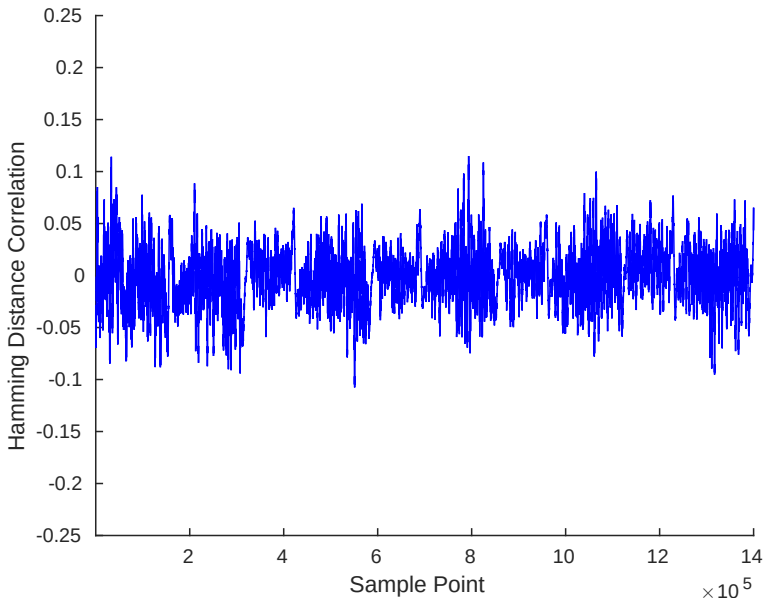

Fig. 5.7: Result of correlating every power sample point to the Hamming distance expected of the SBox1 output for each plaintext

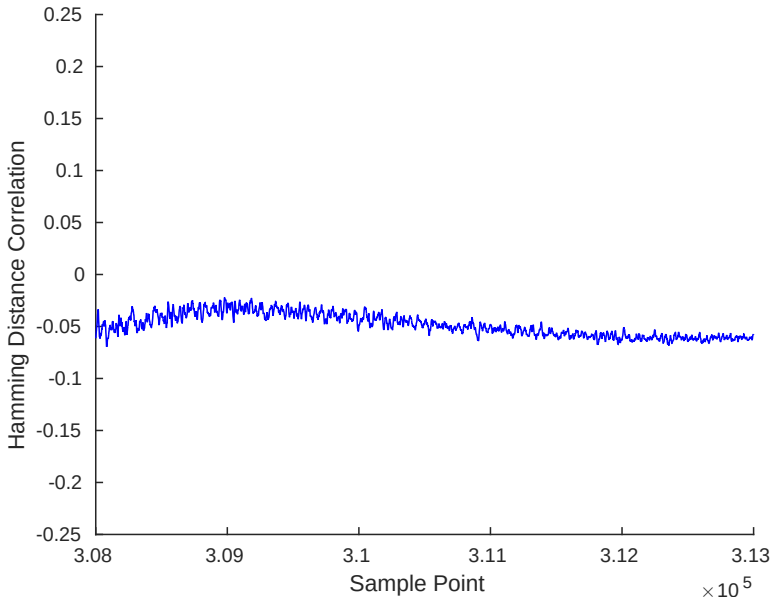

Fig. 5.8: The power correlation for the correct SBox1 input in a window expected to contain the targeted instructions

Since the correlation did not result in a significant spike in the first round, some extra investigation of the DES timing was done. Extra triggers were inserted in the DES implementation just before the targeted instructions. After taking traces with the added triggers, the average difference between the original trigger and the new trigger was used to approximate the location of target instructions in the power trace. In absence of a strong correlation spike, this approximation was used for the further analysis.

5.3 Baseline Result

To obtain baseline CPA results, CPA was performed using all 1000 power traces averaged over all 125 records. The target expected Hamming distance for each key guess was determined, and the correlation between these Hamming distances and the power traces were computed, on a window covering the target instructions power consumption. The 64 key guess correlations for the first SBox are plotted in figure 5.3.

For each guess, the maximum correlation value within the range around the target instruction location is taken as that guess' power correlation value. These values for each guess and each SBox are shown in figure 5.3. For each SBox the guess with the maximum power correlation value is considered the best guess for that part of the key. None of the best guesses matched the actual key used in the encryption.

5.4 Electromagnetic Coupled Results

The same CPA process was repeated for the averaged voltage traces after being filtered by the EM coupling Kalman Filter. The resulting correlation for all 64 key guesses for an SBox is shown in figure 5.4. As before applying the Kalman Filter, no really significant correlation is present. The key guesses, shown in figure 5.4, are made as before taking guess corresponding to the maximum correlation. None of the partial key guesses produced by CPA here are correct.

Because neither of the CPA executions resulted in recovery of all or part of the secret key, the process was not repeated for lesser numbers of averaged records. This did not result in a full examination of the noise reduction of the developed EM coupling as desired. As

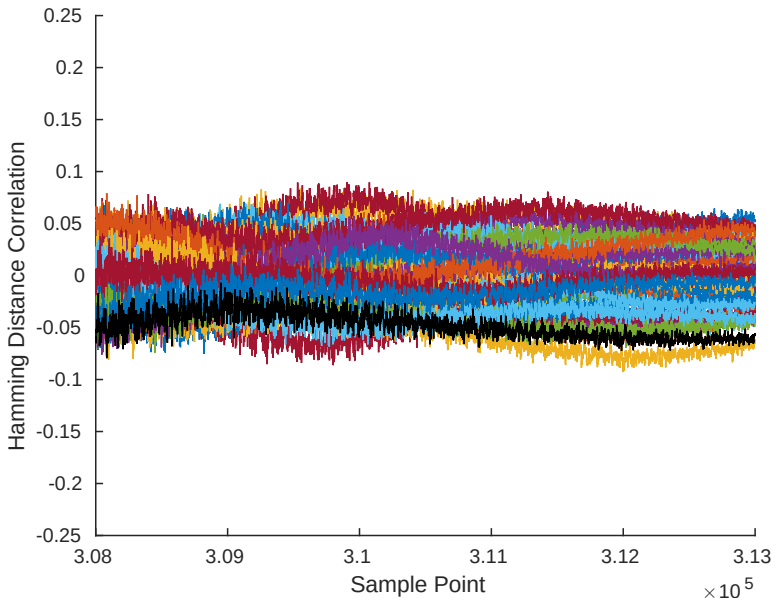

Fig. 5.9: The baseline power correlation within the targeted window for all 64 possible SBox1 subkey guesses

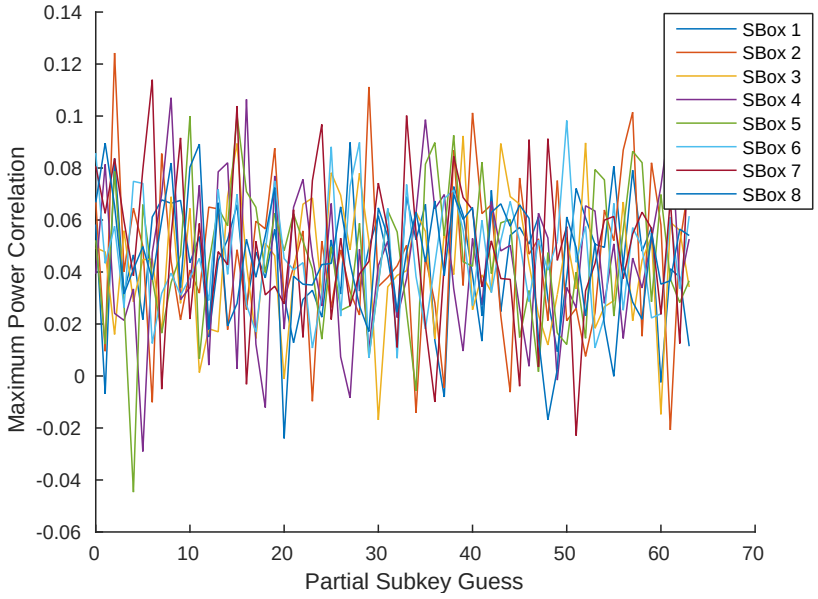

Fig. 5.10: The maximum correlation between Tiva-C power consumption and each key guess for every SBox. The maximum peak for each SBox is taken as the most likely guess

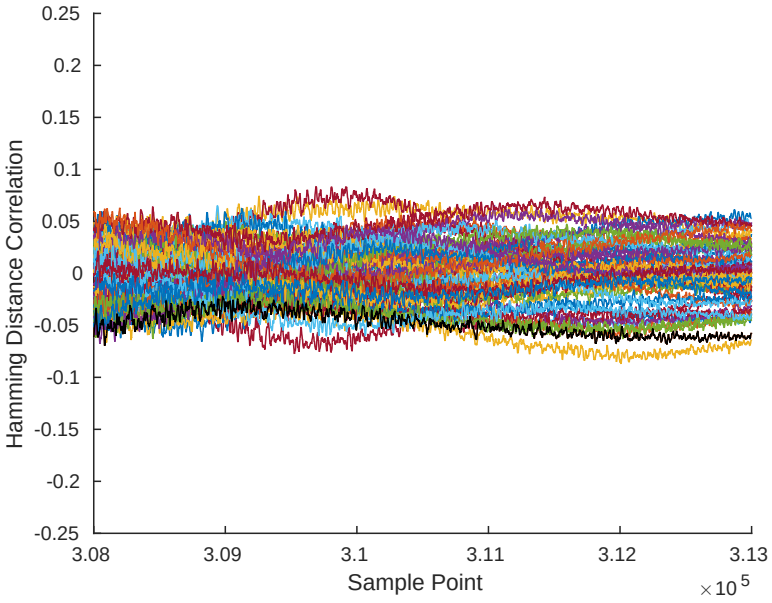

Fig. 5.11: The EM coupled power correlation within the targeted window for all 64 possible SBox1 subkey guesses using EM coupled power traces

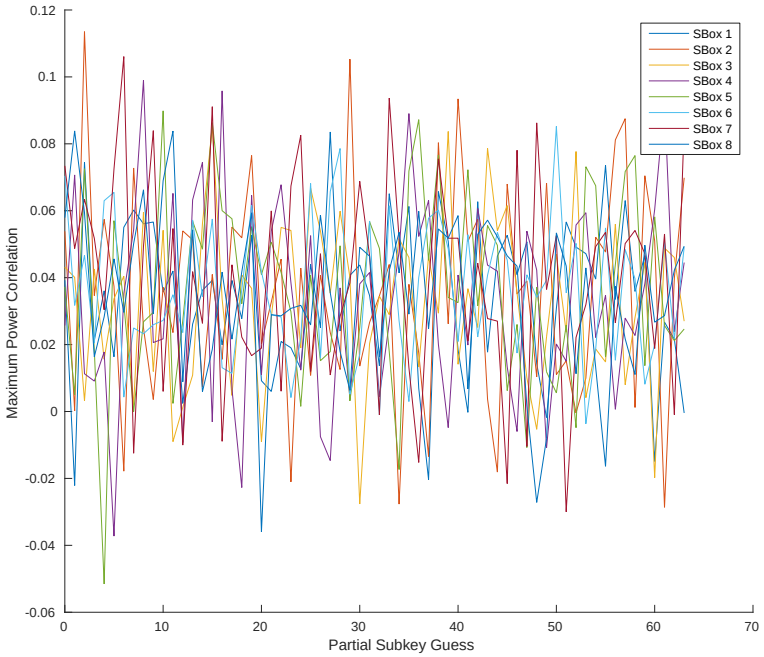

Fig. 5.12: The maximum correlation between Tiva-C power consumption and each key guess for every SBox, using EM coupled power traces. The maximum peak for each SBox is taken as the most likely guess

referred to in section 5.1, the failure of CPA is most likely due to the small magnitude of change in voltage per Hamming distance.

CHAPTER 6

Basys-2 FPGA CPA Implementation

This chapter covers the application of CPA using the Kalman EM coupling method to another crypto-device. For this experiment a Basys-2 FPGA was used to implement and run a DES algorithm. The process of obtaining voltage and EM measurements was repeated, as described in chapter 4, for the FPGA device's power consumption.

6.1 Correlation Power Analysis

The code listing in 6.1 shows the register being targeted for CPA. Every clock cycle the output of the current DES round is loaded into the L and R registers, replacing the input of the current round and becoming the input to the next round. This operation incurs power consumption that is correlated to the Hamming distance between Rout and R. That power consumption is the mechanism by which sensitive information is leaked, and so the CPA implementation is targeted at that operation. Even though all of the bits are being loaded into the same register, each of the SBox outputs are independent, and so these outputs are targeted separately using the Hamming distance of only the four bits corresponding to their output.

For each different plaintext and each potential partial subkey used to generate the power traces of FPGA operation, the Hamming distance for this register load operation, when

```
1 reg [1:32]  L, R;
2
  always @(posedge clk)
4         L <= #1 Lout;

6 always @(posedge clk)
          R <= #1 Rout;
```

Fig. 6.1: The CPA target register as implemented in the FPGA's Verilog code. The register is updated at the end of each DES round, which are completed within one clock cycle.

occurring at the end of DES round one, was determined. Then for each power trace sample point, the correlation between the power consumption and expected Hamming distance was computed, for each subkey guess.

The resulting correlations were compared to determine the most likely subkey guess. For each correlation, corresponding to a particular subkey guess, the maximum value was taken from a range around the targeted register operation. Then for each SBox the most likely guess is determined to be the subkey guess corresponding to the highest resulting correlation value. The experimental results for both the baseline and EM coupling CPA implementations are presented in the following sections.

6.2 Baseline Result

To obtain baseline CPA results, CPA was performed using 1024 power traces, corresponding to different plaintexts, averaged over a total of 256 recorded traces each. The target expected Hamming distance for each key guess was determined, and the correlation between these Hamming distances and the power traces were computed, on a window covering the cycle in which the targeted register load operation occurred. The correlations for the 64 partial subkey guesses corresponding to the first SBox are plotted in figure 6.2 and again in 6.2, expanded to the targeted range.

For each guess, the maximum correlation value within the range around the target instruction location is taken as that guess' power correlation value. These values for each guess and each SBox are shown in figure 6.2. For each SBox the guess with the maximum power correlation value is considered the best guess for that part of the key. None of the best guesses matched the actual key used in the encryption.

6.3 Electromagnetic Coupled Results

The same CPA process was repeated for the averaged voltage traces after being filtered by the EM coupling Kalman Filter. The resulting correlation for the 64 partial subkey guesses for the first SBox is shown in figure 6.3 and again in 6.3, expanded to the targeted range. As before applying the Kalman Filter, no obviously significant correlation peaks are

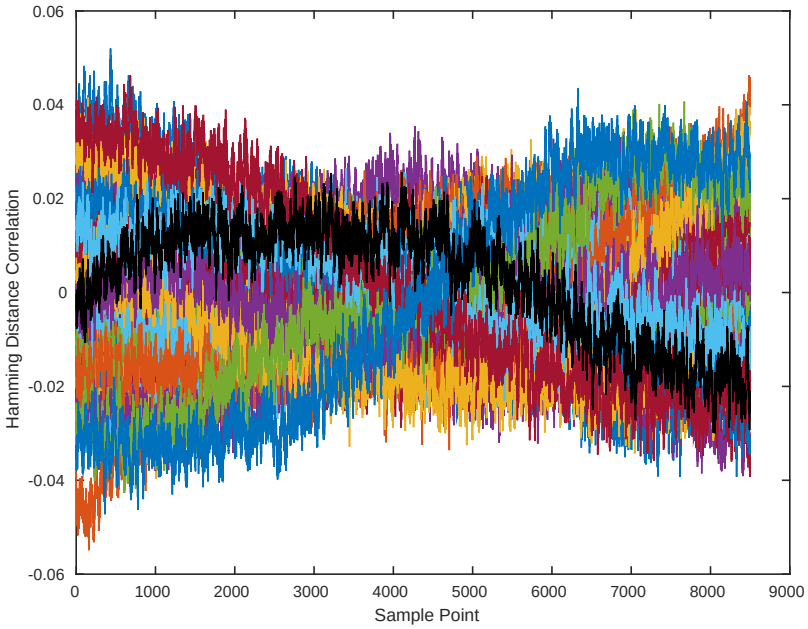

Fig. 6.2: The baseline power correlation for all 64 possible SBox1 subkey guesses

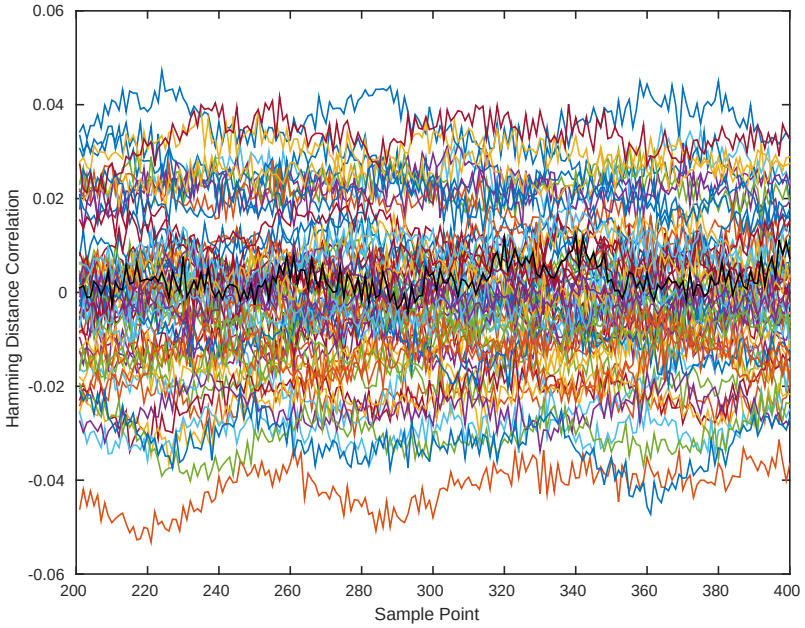

Fig. 6.3: The baseline power correlation expanded to just the targeted window for all 64 possible SBox1 subkey guesses

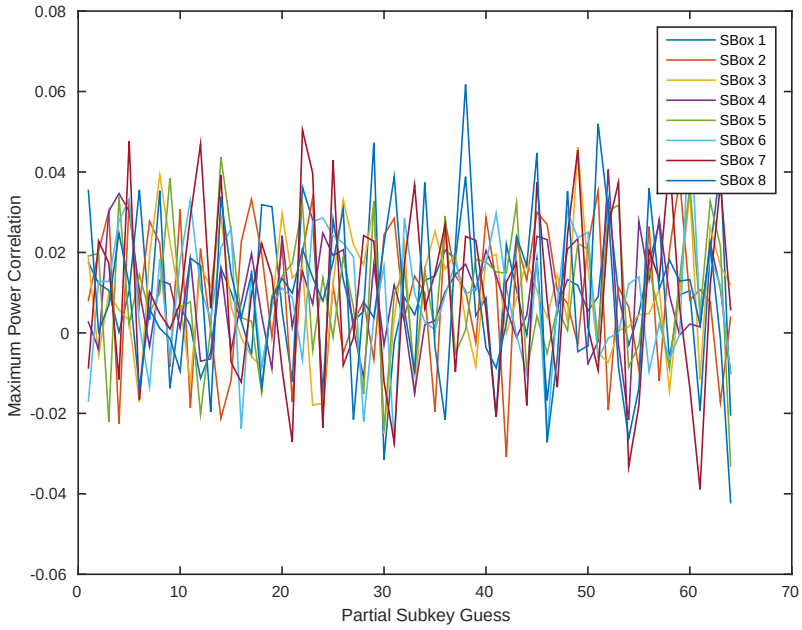

Fig. 6.4: The maximum correlation between FPGA power consumption and each key guess for every SBox. The maximum peak for each SBox is taken as the most likely guess

present. The key guesses, shown in figure 6.3, are made as before taking guess corresponding to the maximum correlation. None of the partial key guesses produced by EM coupled CPA here are correct.

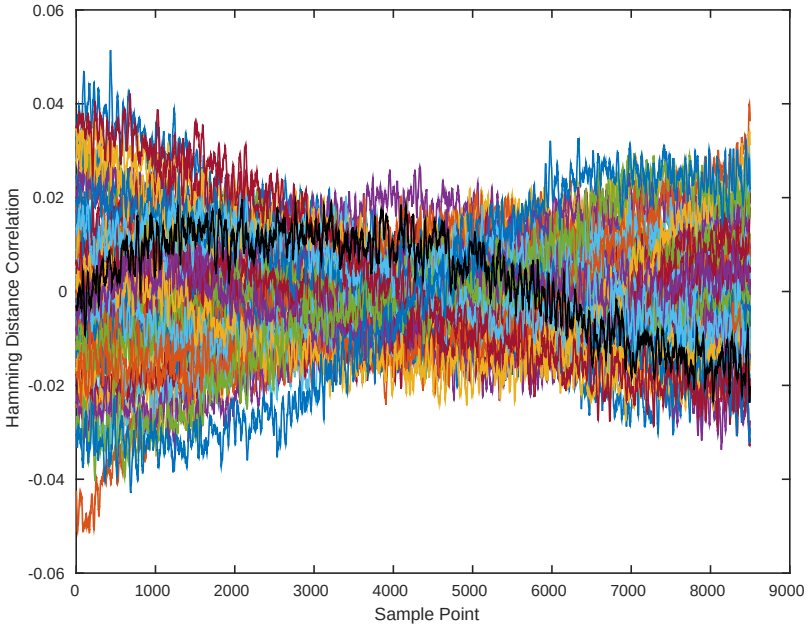

Fig. 6.5: The EM coupled power correlation for all 64 possible SBox1 subkey guesses using EM coupled power traces

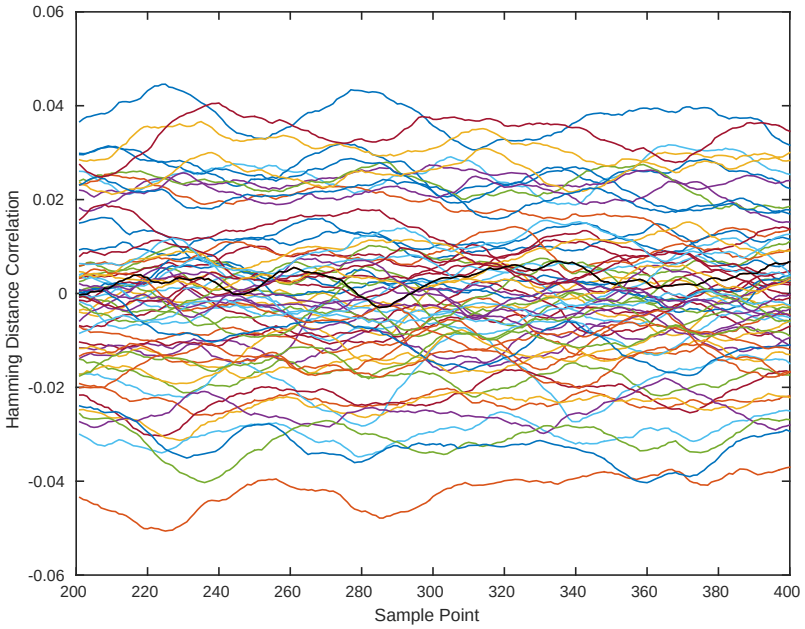

Fig. 6.6: The EM coupled power correlation expanded to just the targeted window for all 64 possible SBox1 subkey guesses using EM coupled power traces

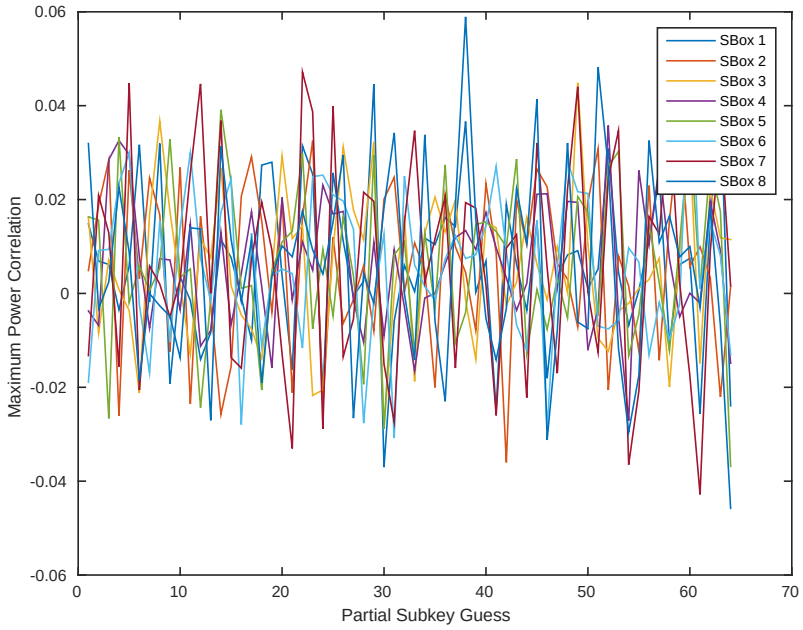

Fig. 6.7: The maximum correlation between FPGA power consumption and each key guess for every SBox, using EM coupled power traces. The maximum peak for each SBox is taken as the most likely guess

CHAPTER 7

Conclusion

The analysis of a Kalman filter based EM coupling did not show to be an improvement to the CPA implementations analyzed. Both CPA implementations, baseline and EM coupled, resulted in the same number of correct partial key guesses for each device CPA was applied to. Also, a reduction of correlation magnitude was observed when comparing the results of the EM coupled to the Baseline implementations. It is hard to tell if this is the result of filtering out extraneous information, the targeted sensitive information, or both but it is likely that a different process variance model could be helpful or necessary to remove unwanted noise while retaining the targeted information.

There were some implementation difficulties that should be taken into account if considering further analysis of this Model. One of which is the small difference in power consumption caused by different Hamming distances combined with limitations of the equipment used to take the records. The voltage difference induced by different Hamming distances were so small that it was unlikely to see any significant differences with the resolution of measurements taken. Another challenge, that presented itself, was the uncertainty of the timing of the targeted instructions. If a more precise timing had been known the power correlation could have been examined more closely. Maybe within a smaller window some level of improvement could have been shown in correlation of the correct key guess, due to filtering.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] C. Z. Liew, R. Shaw, and L. Li, "Protect biometric data with compound chaotic encryption," *Security and Communication Networks*, 2014.

[3] M. E. Smid and D. K. Branstad, "Data encryption standard: past and future," *Proceedings of the IEEE*, vol. 76, no. 5, pp. 550–559, 1988.

[4] W. Diffie, "Exhaustive crypianaiysis of the nbs data encryption siandard," 1977.

[5] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 16–29.

[6] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em sidechannel (s)," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2002, pp. 29–45.

[7] P. FIPS, "46-3: Data encryption standard (des)," *National Institute of Standards and Technology*, vol. 25, no. 10, pp. 1–22, 1999.

[8] B. Yang, K. Wu, and R. Karri, "Scan based side channel attack on dedicated hardware implementations of data encryption standard," in *Test Conference, 2004. Proceedings. ITC 2004. International*. IEEE, 2004, pp. 339–344.

[9] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2001, pp. 251–261.

[10] W. Hnath, "Differential power analysis side-channel attacks in cryptography," Ph.D. dissertation, Worcester Polytechnic Institute, 2010.

[11] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[12] Y. Souissi, S. Guilley, J.-l. Danger, S. Mekki, and G. Duc, "Improvement of power analysis attacks using kalman filter," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2010, pp. 1778–1781.

[13] G. Rigatos and S. Tzafestas, "Extended kalman filtering for fuzzy modelling and multisensor fusion," *Mathematical and computer modelling of dynamical systems*, vol. 13, no. 3, pp. 251–266, 2007.

[14] J. Sasiadek and Q. Wang, "Sensor fusion based on fuzzy kalman filtering for autonomous robot vehicle," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 4. IEEE, 1999, pp. 2970–2975.

[15] E. Foxlin, "Inertial head-tracker sensor fusion by a complementary separate-bias kalman filter," in *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*. IEEE, 1996, pp. 185–194.

[16] Q. Gan and C. J. Harris, "Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion," *IEEE Transactions on Aerospace and Electronic systems*, vol. 37, no. 1, pp. 273–279, 2001.

[17] H. E. Rauch, C. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[18] F. T. Ulaby, E. Michielssen, and U. Ravaioli, "Fundamentals of applied electromagnetics 6e," *Boston, Massachussetts: Prentice Hall*, 2010.

[19] Y. Souissi, J.-L. Danger, S. Mekki, S. Guilley, and M. Nassar, "Techniques for electromagnetic attacks enhancement," in *Design and Technology of Integrated Systems in Nanoscale Era (DTIS), 2010 5th International Conference on*. IEEE, 2010, pp. 1–6.