Side Channel Power Analysis of an Embedded Device Running DES

Justin Cox and Tyler Travis
Department of Electrical and Computer Engineering
Utah State University
Logan, Utah 84322
email: justin.n.cox@gmail.com, tyler.travis@aggiemail.usu.edu

***Abstract*— Hardware security is an ever increasing area of study since exploits have been found on computer systems. Encryption algorithms are very difficult to break. Instead of breaking the encryption algorithm, it is common for an attacker to attempt to recover the encryption key instead. One way of recovering the key is using side channel analysis . This paper will discuss a side channel analysis performed on a microcontroller that is operating as a crypto device. The power traces collected will then be analyzed using Differential Power Analysis (DPA) and the results will be shown and discussed.**

***Index Terms*—encryption, decryption, security, DPA, side channel.**

I. INTRODUCTION

Through side channel analysis, an attacker is able to leak information about a device through natural or physical means. In regards to a device running a crypto algorithm, side channel analysis can be used to leak the encryption key. There a many different kinds of side channel attacks, but this paper will focus on obtaining information from the power consumed by the target device. Different operations and data bits require different amounts of power consumption. By recording and analyzing the power consumption, it is possible to obtain the encryption key. The encryption algorithm used in this paper is the Data Encryption Standard (DES).

A DES algorithm will be programmed and uploaded onto a TI Tiva C microcontroller. The algorithm will be ran for many different plaintext inputs and the power consumption will be recorded using an oscilloscope. The power traces will then be analyzed and it will be determined whether or not the secret encryption key can be obtained.

II. DES

A. Overview and Implementation

The type of encryption programmed on the microcontroller will be DES. Although DES is not the current encryption standard and is not as secure as the Advanced Encryption Standard AES, DES is still used in devices today and side channel analysis of a DES crypto device is a valid security risk.

A brief overview of DES will be given so that the reader has a better understanding of how the algorithm works and will better understand where DPA can be used. If the reader would like an in-depth understanding of DES, it is recommended that the reader look to other sources [1].

The DES algorithm takes a 64-bit plaintext input. It is then run through an initial permutation that outputs 56-bits which are then split into two halves. The data goes through sixteen rounds that each have a sub-key that is generated for each round based on the original 64-bit DES key. After the sixteenth round, the output is run through a finial permutation and the algorithm outputs a 64-bit encrypted ciphertext. The rounds are illustrated in Figure 1.

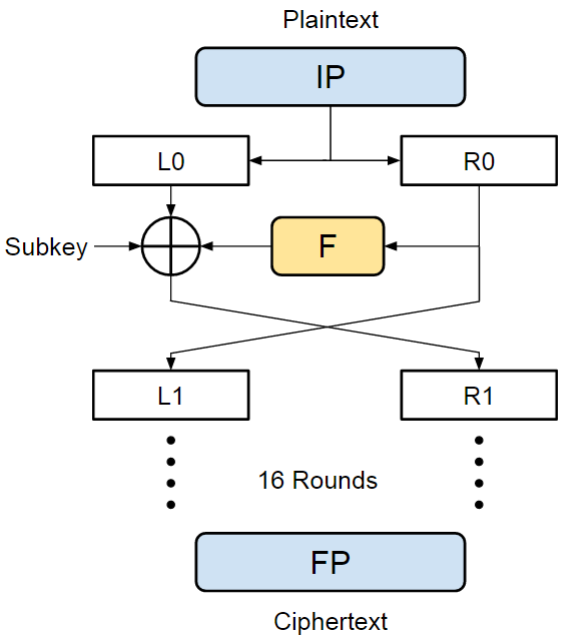

Fig. 1. An illustration of the sixteen round DES algorithm.

During each round, the left 32-bit halve is XORed with the sub key corresponding to the current round as well as the output of the F-function. The inside functionality of the F-function is illustrated in Figure 2. The output of the XOR is used as the next round's right halve and the next round's left halve is the previous round's right halve.

B. Modifications

There were a few minor changes made to the DES algorithm to facilitate the capturing of power traces. These changes do not decrease the security of the DES algorithm. The following assembly routine was added during the last round of the DES algorithm:

```
MOVS    r2,#0x00 ;Set r2 = 0
LDR     r5,[pc,#1012] ;Lower GPIO PIN
LDR     r5,[r5,#0x00]
```

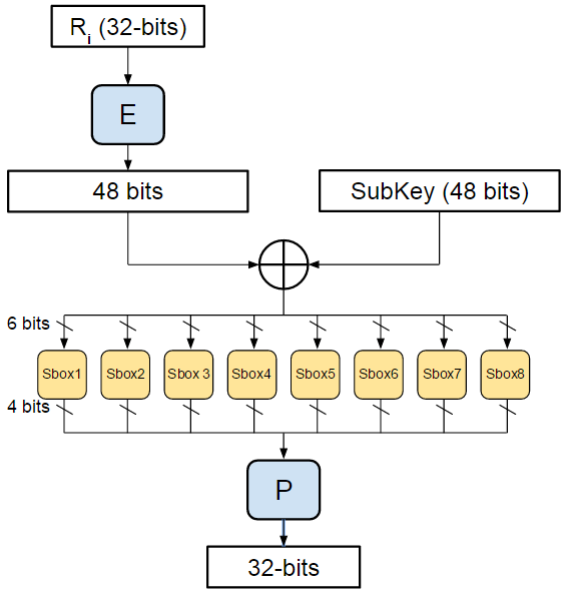

Fig. 2. An illustration of the F-function.

```
BIC     r5,r5,#0x10
LDR     r6,[pc,#1008]
STR     r5,[r6,#0x3FC]

NOP
NOP
NOP
NOP

EOR     r2,r4,r3 ;<--Instruction of Interest

LDR     r5,[pc,#992] ;Set GPIO Pin High
LDR     r5,[r5,#0x3FC]
ORR     r5,r5,#0x10
LDR     r6,[pc,#984]
STR     r5,[r6,#0x3FC]

NOP
NOP
NOP
NOP
```

A GPIO pin on the microcontroller is transitioned from a high to a low in order to trigger the oscilloscope to capture the XOR operation that works with the bits of interest for the DPA. The register that holds the value of the XOR is explicitly set to 0x00 in order to make it easier to measure the Hamming Weight and Distance of that register. This information is helpful for a Correlation Power Analysis (CPA).

III. EXPERIMENTAL SETUP

Acquiring the power traces is the most crucial part of the DES encryption key recovering process. It is important to get as much accurate information as possible. The Tiva C was programmed to run the DES algorithm on 10,000 different plaintext inputs. These 10,000 plaintext inputs would be ran 20 times through DES in order to allow averaging of the traces to mitigate the effects of noise.

A 100 ohm resistor was connected in series with the Tiva C and an external power supply. Two oscilloscope probes where connected across the resistor. This was done so that a differential voltage measurement could be recorded. A third oscilloscope was placed on the output of a GPIO pin that is used to trigger the oscilloscope to capture the relevant data. A picture of the setup is illustrated in Figure 3.

A MATLAB script was used to communicate with the oscilloscope and automate the capturing of the power traces. This was a necessary step because in order to increase the chance of a DES key recovery, a large number of power traces need to be recorded. The automated capturing sequence takes about 1 second to record 1 power trace. Since we have a total of 200,000 power traces, the total time required to record all of the traces was around 55.5 hours.

The oscilloscope that was used was able to sample at a rate of 2.5 GHz. The Tiva C was programmed with an internal clock of 16 MHz. The oscilloscope was configured to record 10,000 sample points per capture. There were two different modes used to capture the power traces. One round of power traces were captured using the default mode of 'Sample'. The other set of power traces were captured using the 'High Res' mode. The results of each mode will be discussed in a later section of the paper.

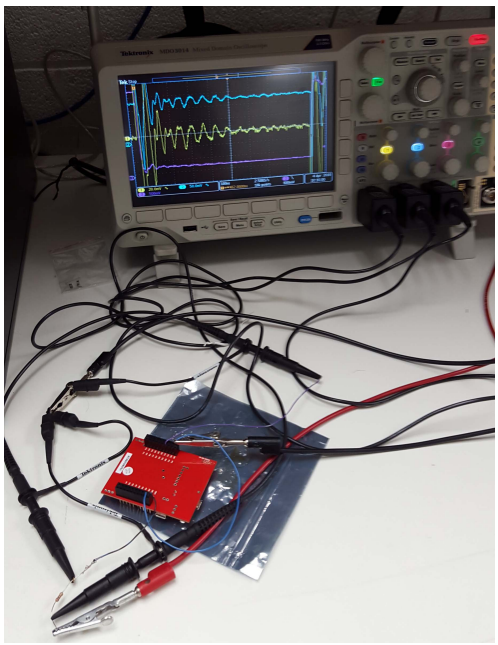

Fig. 3. A picture taken of the experimental setup used.

IV. POWER TRACE ANALYSIS

A. DPA

The power traces were uploaded into MATLAB and DPA was used to analysis the data. There are papers which discuss DPA on the DES algorithm and the authors of this paper perform a similar method [2].

In order to perform the analysis, a DES round must be chosen. Since the plaintext values and the cyphertext values are known, it is recommended to choose the first or last round of the DES algorithm. This paper has chosen to analysis the last round of the DES algorithm. In order to recover the key, an operation that contains the data from the subkeys must be chosen. This paper will focus on the last round's XOR operation that takes the F-function's output and the left halve of the cyphertext output as its inputs.

The bits from the previous round's input that are XORed with the key bits will be referred to as D-bits. We would like to find out what the D-bits are. We know the output of the XOR operation from the cypertext. We can guess the 48-bit subkey 6 bits at a time because each 6-bit chunk is sent into different S-boxes. Guessing 6-bits is not that intensive because there are only 64 possible values. We then XOR the output of the F-function which contains our key guesses with the cyphertext to get the D-bit values. Then for each Sbox we check the corresponding bits.

If the bit is "1" it is put into a subset A1. If the bit is "0" it is put into a subset A0. The subsets are then averaged across all sample points. The difference of A1 and A0 is taken to give us a waveform for each key guess of each Sbox. If the key guess is correct, there will be greater power consumption compared to the incorrect key guesses.

In order to see a greater change in power consumption, four bits of the D-bit value can be compared instead of just one. The traces are then organized into two subsets, A1 and A0, where A1 has traces that correspond to D-bits "1111" and A0 has traces that correspond to D-bits "0000". All of the other combinations of D-bits will be ignored. This is the main disadvantage to using more than one bit.

B. CPA

Since the result of the XOR operation is stored in a register with a previous value of 0x00, the Hamming Distance and Weight can be calculated. This makes a key recovery using CPA possible. However, the authors did not have enough time to try CPA in addition to DPA.

V. RESULTS

The plots generated in MATLAB are shown in the Appendix. Since the actual encryption key was known, the correct guess for each Sbox has been emphasized in black. Overall, entire 48-bit subkey was not able to be recovered using DPA on the power traces recorded.

A. DPA 1-bit

For some of the Sboxes, it is possible to limit the 64 key guesses down to a couple of guesses. However, the correct key guess is hidden for the remaining Sboxes. It can be seen that the High-Resolution traces yielded better results.

B. DPA 4-bit

For the 4-bit DPA implementation, there is only one Sbox that the correct key could be limited to a couple of guesses. Once again, the High-Resolution traces yield a better result.

VI. CONCLUSION

REFERENCES

[1] G. Edward Suh, C. W. O'Donnel, I. Sachdev, and S Devadas. Design and Implementation of the AEGIS Single-Chip Secure Processor Using Physical Random Functions. *Proceedings of the 32nd annual international symposium on Computer Architecture*, 2005.

[2] M. Deutschman, "Cryptographic Applications with Physically Unclonable Functions," M.S. Thesis, Inst. Mathematics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria, 2010.