# Side-Channel Attacks

Aleksei Ivanov

26th November 2005

## 1 Introduction

Side-channel attacks are described in [5] as follows. Side-channel attacks are attacks that are based on side channel information. Side channel Information that can be retrieved from the encryption device that is neither plain text to be encrypted nor the cipher text resulting from the encryption process. There are several kind of side-channel attacks, in the [2] timing attack is referred to as the most common one. Then There is one kind of information leakage referred to in [1] as power consumption leakage, that is a big help to the timing attacks. It is even harder to protect a system against the power consumption attacks when attacker has direct access to the encryption device.

The purpose of this paper is to get an overview of attacks on encryption systems where an attacker is using other ways to obtain the encryption key than breaking the mathematical algorithm. For instance the attacker can measure the time of computation to determine the complexity of the operation. Thus having some information about the encryption algorithm the attacker can guess what operations might have occurred. For instance there is a an operation that takes less time to compute when given certain input. This type of attack is called timing attack. There are more ways to find out the key with help of some information leakage from devices.

## 2 Side Channel Attacks

### 2.1 Timing Attacks

This type of attacks are based on time it takes to do an operation. Attacker can passively listen on the channel and record every input, output and the consumed time. In some implementations multiplication by zero bypasses the multiplication process and patches the result with all `0's`. This results in small, constant time when compared to complete multiplication. Timing side channel information can be obtained either by precisely measuring the time taken by one encryption or by averaging the time taken over several encryptions [4].

### 2.2 Power Consumption Attacks

Devices consume power and the power dissipated by a device is an other side channel. Differential power analysis (DPA) is a power consumption side channel attack that divides the encryption into a number of time slots and measures power in each slot for different plain text input. A small number of the power measurements correlate with each bit of the interval stage during encryption [4].

This attack requires little knowledge of the device and is difficult to hide the channel information if the attacker has direct contact to the device.

### 2.3 Error Message Attacks

It is important to be careful with sending out information about operation failures. Because an attacker can use it to compute decryption of any cypher text by using the device as an oracle as referred to in [1]. The attack is theoretically possible against RSA encryption standard PKCS#1. Attacker sends well chosen "cipher texts" to the device and using it as an oracle to know if the corresponding plain text is in the right format. The attack is reliable if the different failures can be separated.

#### 2.3.1 Fault Attacks

Results output by a faulty implementation is another side channel. Fault-based side channel attacks are based on the observation that errors de-

liberately introduced into a device and the resulting faulty computations leak information about the implemented algorithm [4].

Boneh, DeMillo and Lipton [6] presented the first fault-based side channel attack against devices implementing public key cryptography. They showed that faults could be introduced at a random bit location in one of the registers in the encryption device by exposing it to ionizing and microwave radiations. They showed how information leaking from such a fault-based side channel could be exploited to factor the modulus of an RSA implementation.

When attacker has physical access to a cryptographic device the attacker may try to make the device malfunction. This kind of attack is based on obtaining about the message or the secret key from the output of erroneous computations. The following descriptions are based on [1].

There are several ways to make an error occur in the device. Here are some non-invasive methods:

- spike attacks work by applying more power to the device than the device can handle.
- glitch attacks target the clock contact in the similar way to spike attacks
- optical attacks work by applying light to the device in order to alter some bits.

Depending on where attacker is able to do the attack models are divided into categories:

- control on the fault location;
- control on the fault occurrence time
- control on the number of faulty bits induced
- the fault model

#### 2.3.2 Differential Fault Analysis (DFA) attacks

Biham and Shamir [7] presented a fault-based side channel cryptanalysis of DES called Differential Fault Analysis (DFA). They showed how DFA can find the last DES round key using less than 200 cipher texts by assuming that one bit of data in one of the 16 rounds is flipped with a uniform probability. Floyd, Fu and Sun [8] presented a similar DFA attack on RC5 by introducing the register faults that affect the current round and then comparing the faulty result with the correct one to obtain the round key. They found the round key of last round first, and then the round key of the round before last, and so on. Biham and Shamir extended their fault model to show that DFA can uncover the structure of an unknown cryptosystem implemented in the encryption device [4].

### 2.4 Frequency-based Side Channel Attack

In [3] chapter 4 there is an experiment with an *ARM Integrator/C7TDMI* core. In this experiment the researcher tries to measure power consumption and electromagnetic emanation. Since the core is assumed to consume little power and have small size, it is frequently used in embedded devices such as pagers, wireless handsets, and personal digital assistants (PDA).

#### 2.4.1 Experiment Set-up

The experiment set-up is described as follows. A digital oscilloscope, an EM probe connected to a pre-amplifier, a Multi-ICE debugger, a personal computer, and a inductive probe are used to acquire both EM and power traces from the ARM Integrator/C7TDMI core module. See Figure 1.

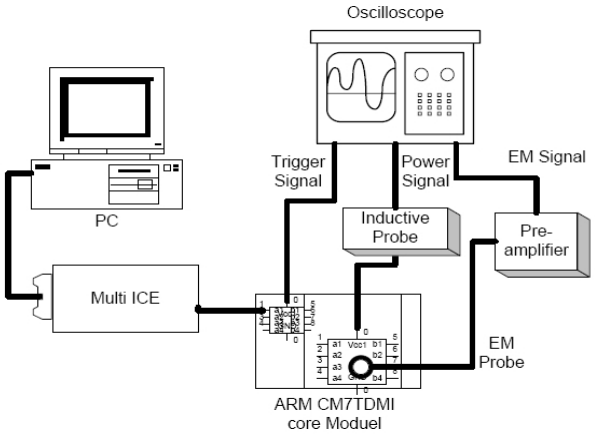


Figure 1: Power and EM Measurement Set-up on ARM Integrator/C7TDMI

To measure EM and power traces, a trigger signal is needed to notify the oscilloscope when to start recording a trace. In the experiments on the ARM

core module, the trigger signal is sent in the form of a software interrupt from the core module to the oscilloscope. For all traces measured for the ARM experiments, the HiRes acquisition mode is used. In HiRes mode, the instrument creates a record point by averaging all samples taken during an acquisition interval. It results in a higher resolution, less noise, and lower bandwidth waveform. To eliminate noise, averaging of at least several hundred of frames is required. The number of data points in each trace is specified by the record length and the number of traces is specified by the frame count. Each frame is captured when a valid trigger occurs.

For instance, an AES encryption algorithm is too long to fit into one frame. In most cases, the adversary is only interested in a small section of the Rijndael encryption algorithm, particularly the output of the S-Box in 1 round of AES. Therefore, it is preferable to pinpoint the attack point by zooming into the section of interest using the delay mode on the oscilloscope. The delay mode allows the oscilloscope to start displaying waveform by a user-specified period after the start of the trigger signal. For instance, if the specified delay period is 1 ms, the scope will display a waveform 1 ms after the start of the trigger signal. As a result, the attacker can pinpoint the exact section that contains the load instruction at the output of the S-Box. Hence, this allows the attacker to increase the frame resolution and capture only the desired section into a single frame.

#### 2.4.2 Experiment methodology

The experiment is carried out in three steps. Step one, Loading the AES Encryption Program to the ARM Evaluation Board. Step two, Capturing EM or Power Traces. And step three, Statistical Analysis with MATLAB.

*Step one.* The experiment is conducted on AES encryption with 128 bit key length. The test Program runs the encryption many times with random plain text input. the input is kept for statistical analysis.

*Step two.* The EM probe is placed on top of the core processor before running the encryption algorithm. Then the oscilloscope is set up the way instructed in 2.4.1 and then the encryption algorithm is run. After the process the data from oscilloscope is exported into MATLAB.

*Step three.* Statistical analysis is run on raw EM data with MATLAB program. The analysis program is written according to DFA attack methodology to produce a correct key guess after running through all possible keys.

#### 2.4.3 Results

*Simple electromagnetic analysis (SEMA)* can show some information about operations the device is performing or key material. Figure 2 shows the scope capture of an AES encryption with 192-bit key length of a single EM frame.

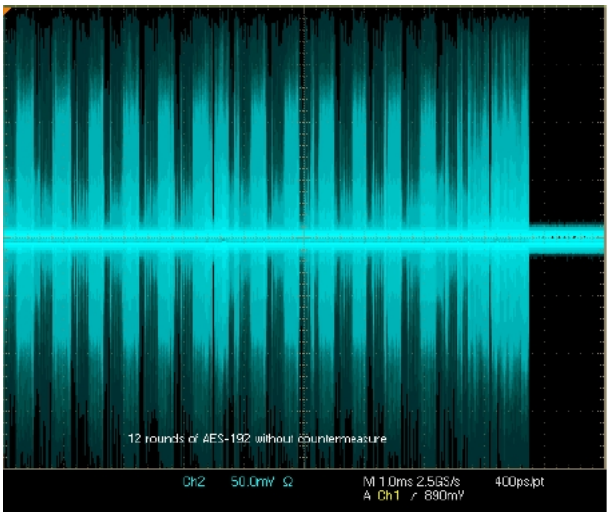


Figure 2: SEMA on PDA for AES 192-bit

There is clearly shown 12 AES transformations. Thus an attacker can determine the key length by inspecting the number of rounds being executed during AES encryption.

# 3 Preventing Side-Channel Attacks

## 3.1 Countermeasures Against Timing Attacks

A counter measure against timing attack, that takes in consideration the property of multiplication by 0 being faster than other multiplication operations, is proposed in [4] to make the time consumed by any operation independent of the input.

A other counter measure would be modify the computation to blind the attacker from the details of the modification. In elliptic-curve public key cryptography computing a multiple $K$ of a curve point $P$ is a common operation that can be blinded from the attacker by pre-selecting a random curve point $Q$, pre-computing $K \times Q$ and computing $K \times P$ as $K \times (P + Q) - K \times Q$.

### 3.2 Countermeasures Against Power Analysis Attacks

A counter measure against DPA is masking the side channel power information by performing random calculations so as to increase the measurement noise. However this random noise is easy to remove, as it tends to average out over time.

Another solution is to add complementary circuits to mirror the real encryption calculations in the device. For instance, if the real circuit is multiplying by the binary number 101, then the complimentary circuit multiplies by 010. This smooths out the over all power consumption since the power consumed by both parts together is approximately constant. Still, since mirroring is not perfect, it is unclear whether all information can be blocked by this solution [4].

A third solution is to vary the order of the operations to make it more difficult for the attacker to identify patterns in power consumption [4].

### 3.3 Countermeasures Against Fault Message Attacks

#### 3.3.1 Fault-based side channel cryptanalysis tolerant architectures

A possible solution against DFA is to do the encryption twice and output the results only if the two outputs are identical. The problem with this approach is, that it doubles the computation time. In addition the probability of a fault occurring twice is not sufficiently small [5].

[4] introduces encryption architectures that can tolerate fault-based side channel analysis. These architectures are called *Concurrent Error Detection (CED)*. The idea is to do encryption and decryption at the same time and check the result before sending it to output. The architecture allows to do this on the algorithm level (shown on figure 3), on the round level (shown on figure 4) and on the operational level (shown on figure 5).

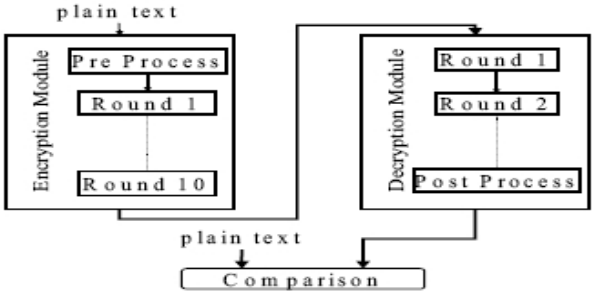


Figure 3: Rijndael encryption with algorithm-level CED

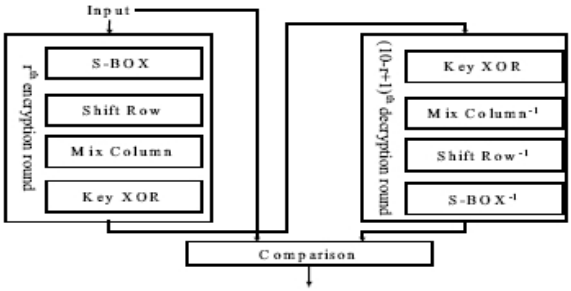


Figure 4: Rijndael encryption with round-level CED

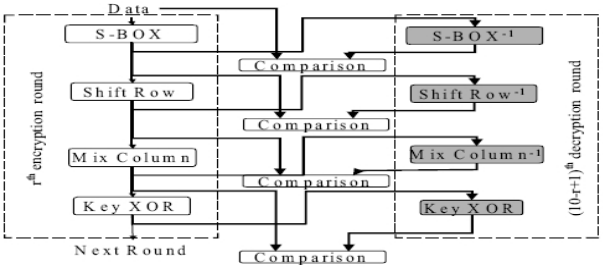


Figure 5: Rijndael encryption with operation-level CED

## 4 Conclusion

There is little research done on side channel information leakage. The results imply that it is possible to obtain encryption keys without breaking

the mathematical algorithm. But results also imply that there have been little publications on attacks on servers or PC-s. Smart card and other simple mobile solutions have the biggest risk of being attacked via side channel. Their simple infrastructure and easy physical access make them easy target for an attacker.

Physically isolated systems, like on-line servers, are usually prone only to timing attacks. This kind of attack is easily shielded with adding delays or altering the algorithm to do its operations in constant time. It may take some additional resource but it is possible to minimize the risk of leaking significant side channel information from the device.

Finding countermeasures for side channel attacks should be a high priority issue in smart card design in the future. People are using smart cards in their everyday life and their well-being is depending on them. Hence the implementations have to be safe. In this case mathematical algorithm alone is not sufficient. Additional countermeasures have to be taken into considerations. There does not seem to be a perfect solution yet since smart cards are not safe from complex differentiated attacks.

# References


[1] URL: https://www.cosic.esat.kuleuven.be/nessie/Bookv015.pdf Pages 344-354

[2] Niels Ferguson, Bruce Schneier, Practical Cryptography. 2003 pages 150-151 and 288-290

[3] Chin Chi Tiu, A New Frequency-Based Side Channel Attack for Embedded Systems. URL: http://optimal.vlsi.uwaterloo.ca/NEW/thesis_AgnesTiu.pdf See Chapter 4

[4] Ramesh Karri, Kaijie Wu, Piyush Mishra Yongkook Kim. Fault-Based Side-Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture. URL: http://www.ece.mtu.edu/ee/faculty/mishra/Publications/DFT2002/DFT2002_Paper.pdf

[5] Hagai Bar-El, Whitepaper on Introduction to Side Channel Attacks. See also http://www.discretix.com/PDF/Introduction%20to%20Side%20Channel%20Attacks.pdf

[6] D. Boneh, R. DeMillo, and R. Lipton, "On the importance of checking cryptographic protocols for faults",Proceedings of Eurocrypt, Lecture Notes in Computer Science, Vol. 1233, Springer-Verlag, pp. 37-51, 1997.

[7] E. Biham, A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems", Proceedings of Crypto'97, 1997.

[8] J. J. Floyd, K.E. Fu, P. Sun, MIT, "6.857 Computer & Network Security Final Project: Differential Fault Analysis", December 1996. http://web.mit.edu/jered/www/publications/rc5-dfa-paper.ps