PROJECT SPECIFICATION – SIDE CHANNEL ATTACKS

Authors:
Ondrej Blazek
Florian Payerl

Advanced Encryption Standard Cipher (AES)

Advanced Encryption Standard is a commonly used block cipher algorithm.
It uses 128 Bit input, called the “state” stored in a 4x4 matrix
The AES Algorithm Consists of four transformation steps called *Sub Bytes*, Shift *Rows*, *Mix Columns* and *Add Round Key*. Figure 1 shows a top level view of the Algorithm including the four steps. In the following paragraphs each of this steps will be explaind.

Sub Bytes (SB):
Sub Bytes uses a substitution table (S-Box) consisting of several substitution values. In the Sub Bytes Step each byte of the state will be replaced with a byte from the S-Box. This provides a monalaphabetic encryption.

Shift Rows (SR):
The rows of the state are left-shifted a specific amount of bytes:
1. Row: none,
2. Row: 1 byte ,
3. Row: 2 bytes,
4. Row: 3 bytes

Mix Columns (MC):
This step mixes the Data in the Rows via a multiplication of the bytes a_j with a constant (either 1,2 or 3).

$$b_0 = (a_0 \cdot 2) \oplus (a_1 \cdot 3) \oplus (a_2 \cdot 1) \oplus (a_3 \cdot 1)$$
$$b_1 = (a_0 \cdot 1) \oplus (a_1 \cdot 2) \oplus (a_2 \cdot 3) \oplus (a_3 \cdot 1)$$
$$b_2 = (a_0 \cdot 1) \oplus (a_1 \cdot 1) \oplus (a_2 \cdot 2) \oplus (a_3 \cdot 3)$$
$$b_3 = (a_0 \cdot 3) \oplus (a_1 \cdot 1) \oplus (a_2 \cdot 1) \oplus (a_3 \cdot 2)$$

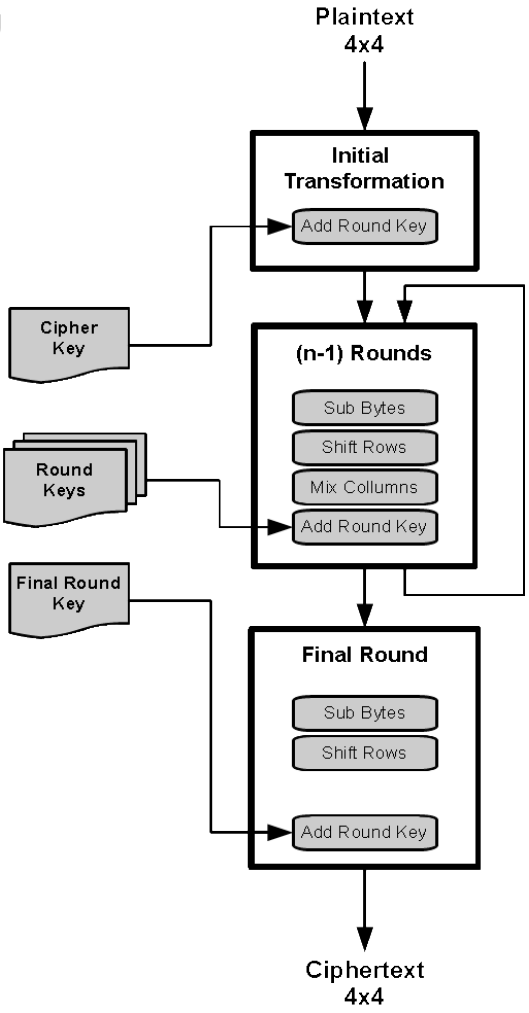

Image 1: AES algorithm

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Add Round Key (RK):

In the Initial Transformation and after each Round a so called *KeyAddition* takes place. This is the only step that depends on the Cipher Key. It generates a new Round key for every Round from the old Round Key.

Side channel attacks based on DPA

Power analysis attack is one of the most common side-channel attack and one of the most powerfull. The attack utilizes information gain from power consumption of the cryptographic system. The goal is to find a relationship between the two states of the system. Between secret key information and the instantaneous power consumption on a bit level. The think is, if we know in what state (in which point of the encryption operation) the system is and we have access to the power consumption, we can very well use that for the attack, because the power consumption can be related to the bit value being manipulated at that time.

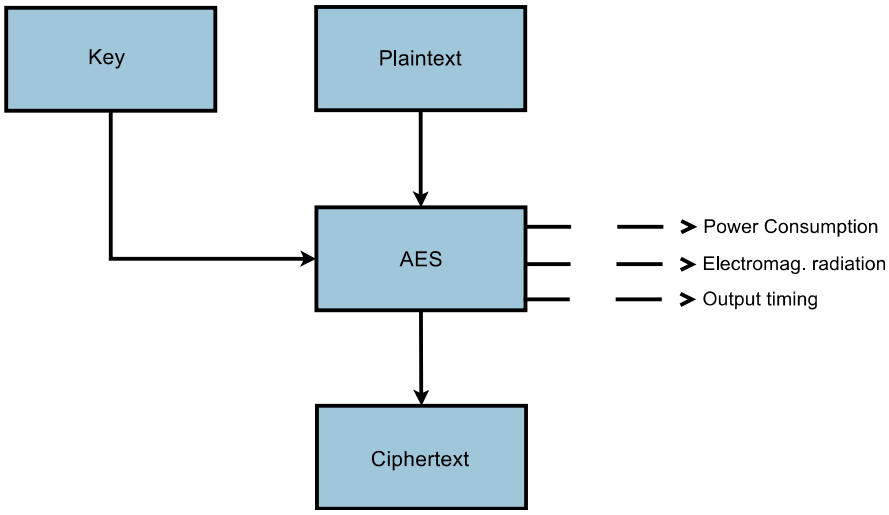

Image 2: Encryption mechanism and side-channel leakages

Hamming weight model says that more power is consumed the more bits are equal to logic 1 and also the number of logic transitions during encryption process is proportional to power consumption.

Also there is basically no way how to detect these kind of attacks, because the monitoring of the device is usually passive.

Differential Power Analysis

The thing is the power variations which are related to the data values being manipulated can be hidden by errors in measurement or noise. However that is possible to overcome it by using differential power analysis and its statistical techniques. So the most important step is to get as many as possible power measurements processed during each operation of the cryptographic system. The next step is to divide the power measurements into subsets which are then checked for differences. So after the capturing (getting) the ciphertext values at a known step and getting the power measurements the next step is to try and guess the key. We need to calculate the value of the bit along with the key guess.

Therefore if we implement this on AES algorithm it goes as follows. The S-box substitution in the SubBytes, which is one part of AES operations, is done on every byte separately. So the attack can be done on one byte at first and afterwards repeated 16 times, which will give us, using the power traces to whole 128bit key. So that distinctly reduces the possible combinations. So the plaintext is unknown and also the cipher key, we then have to choose a bit which is going to be analyzed in the choosen byte location. Since we can capture the power consumption along with each ciphertext value, we can then perform partitioning and analysing.

Now we can go onto the guessing the values of the choosen byte by inverting the SubBytes and XORing it with the partial key guess. Then we will have to partition the power traces into two subsets which will be equal to the computed values of XORing the bit guess with the key guess after inverting the SubByte opeartion. One of the subsets will be equal to logic 0 and the other one equal to logic 1. Then the average of the subsets is computed.

Afterwards these two average values (traces) are examined and depending on a selected bit value, there will be correlation/spike or small noise in case the guess was wrong.

Those steps are then repeated for the rest of the bytes.

Side channel attacks based on DFA

Differential Fault Analysis attacks extract the key by analizing between correct and faulty ciphertexts. A faulty ciphertext can be obtained from fault injection during execution of the algorithm.

The important thing for us in attacking AES is that the last round (the 10^{th}) is not included in the “loop” like the other 9 rounds and the 9^{th} round ciphertext is basically XORed with the final round key value. If we could somehow determine the final round key value, the initial cipher key could then be computed by inverting all the operations.

Simple fault attack

This attack is based on flipping or injecting a fault bit after execution of the 9^{th} round of AES encryption system which in case the system is working correctly will skip the last round and return faulty output by a round reduction fault. That is pretty much the result of the 9^{th} round.

There is a obvious relationship between correct C and faulty ciphertext C' as you can see in

this equation:

$$C = (SR(SB(M^9))) \oplus RK^{10} = (SR(SB(C'))) \oplus RK^{10}$$

Moreover we can substitute the RK^{10} simply with:

$$RK^{10} = (SR(SB(C'))) \oplus C$$

So if we can somehow get ciphertext after 9 rounds along with ciphertext where the bit was changed, we should be able to compute back all steps and finally deduce the cipher key.

REFERENCES

[1] http://people.rit.edu/kjm5923/DPA_attacks_on_AES.pdf

[2] https://www.iacr.org/archive/ches2006/08/08.pdf

[3] http://link.springer.com/article/10.1007%2Fs13389-011-0006-y

[4] https://en.wikipedia.org/wiki/Advanced_Encryption_Standard