Secure Processor MicroArchitecture
Assignment 3

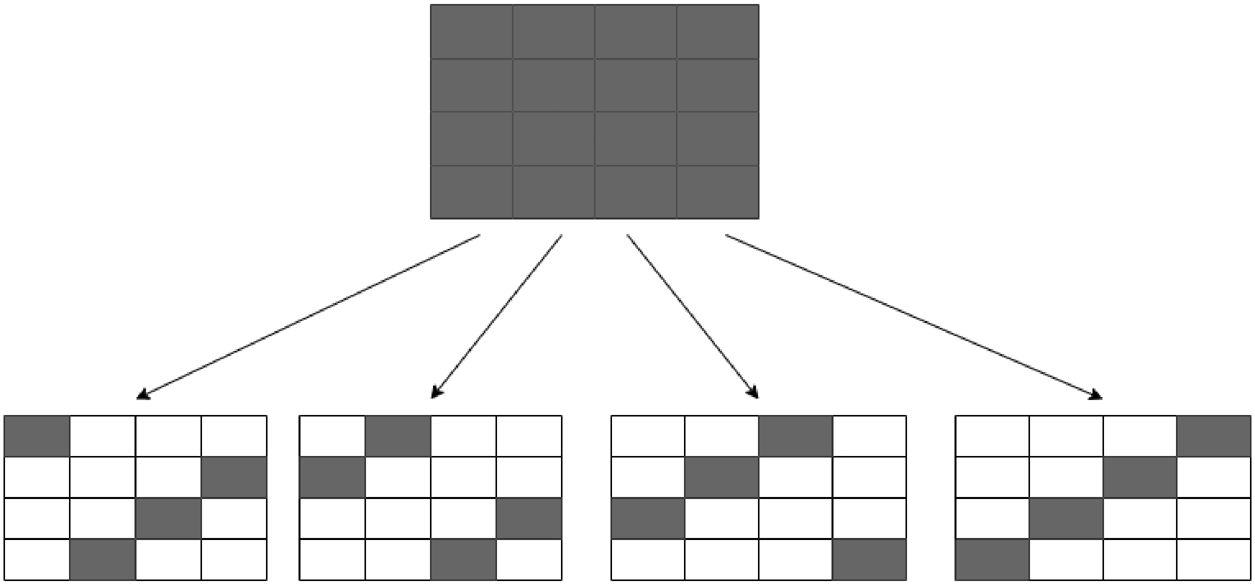
Team Members:
Surya Prasad S - EE19B121
Sai Dheeraj Ettamsetty - CS18B055

1. The first round key we were able to obtain was the 10th round key. With this key, we were able to find all the other round keys and the main key.
2. After offline analysis, we were able to retrieve all the 16 key bytes.
3. Python script has been attached. It generates a file roundkeys.txt which contains all the round keys. Usage is as follows:
   ```
   $ python3 dfa_8.py
   ```
4. **Report**

A. Steps used for the analysis

We are given 4 pairs of ciphertext and faulty ciphertext when a single fault is injected in the 8th round. We don't require all of these pairs and only require 2. For each of these pairs, we have generated 4 pairs of ciphertext and faulty ciphertext as if 4 faults are injected in the 9th round, each fault in a different column.

Faulty cipher text when a byte fault is injected at start of 8th round



when fault is in Column 1 | when fault is in Column 2 | when fault is in Column 3 | when fault is in Column 4

Faulty cipher texts when the faults are injected in different columns at start of 9th round

The white bytes are those bytes which have been replaced with the correct byte (corresponding to the correct ciphertext) and gray bytes are retained from the faulty ciphertext.

Using this method, we have generated 4 pairs of ciphertext-faulted ciphertext pairs from every initial pair we are given. So, we ended up having 16 pairs of ciphertext and faulty ciphertext.
If we use all initial 4 pairs then we feed these 16 pairs as input to the DFA code for 9th round analysis because now each of these ciphertext and faulty ciphertext pairs are no different from the ones generated by injecting the fault in the 9th round.

It is possible that there may be more than one intersection but in our case only 1 intersection was there which is our 10th round key. Once we have the 10th round key, we will give it to the *reverse_key()* function to get all other round keys along with the secret key. From that we also analyzed if our identification of key is right by comparing the output of each round while decrypting both correct and faulty ciphertext. Following is the output of the 8th round while decrypting for the first initial pair of ciphertext-faulty ciphertext.

Output of 8th round while decrypting

For correct ciphertext - 51f02ceffe80d605**81**ecdb72862b3a2b
For faulty ciphertext - 51f02ceffe80d605**2a**ecdb72862b3a2b

As we can see, only one byte is incorrect which is as expected. Hence our analysis was correct and the identified key is verified. This is also the case with other pairs.

B. Time taken reports: Here the least time required which is using 2 initial pairs is given

Time taken to compute 10th round key 3.45175s
Time taken for the whole process 3.45183s

C. We need a minimum of 2 pairs of ciphertext-faulty ciphertexts to retrieve all the bytes of the key