Deep Learning for Side Channel Attack

Group 19

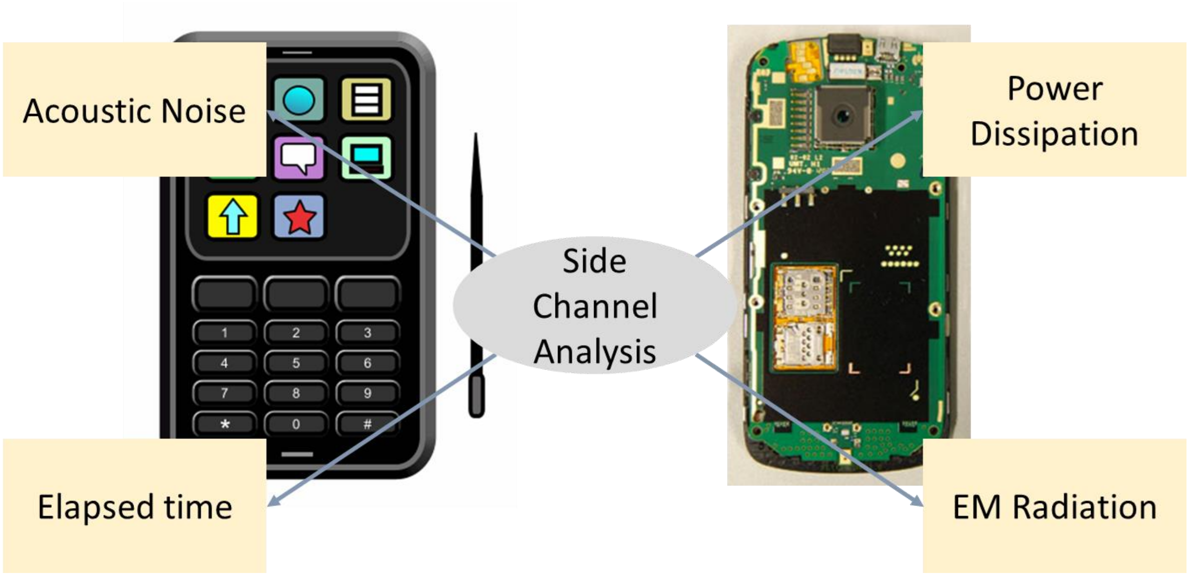
E/17/038 Anurudda
E/17/101 Anjalee
E/17/292 Rilwan

Contents

PROBLEM & SOLUTION

What is SCA?

- Attack that exploits information leaked through the physical implementation

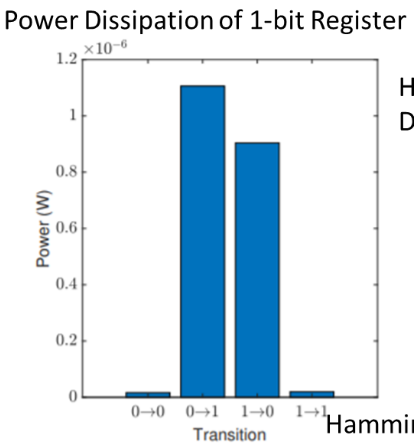


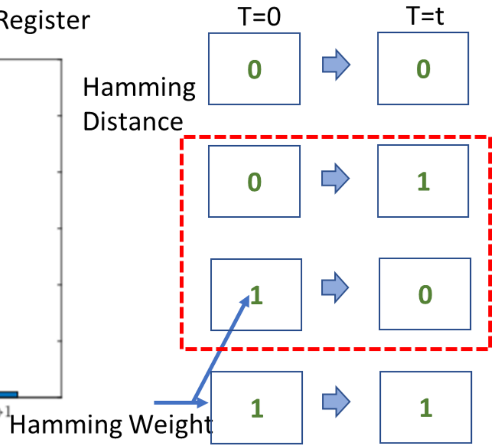
Side Channel Attacks

- Power Analysis Attacks
- Differential Power Analysis (DPA)
- Simple Power Analysis (SPA)
- Timing Attacks
- Electromagnetic Radiation Analysis (e.g., Van Eck phreaking)
- Acoustic Cryptanalysis

Power Analysis Attack

- Revealing the secret information via the power dissipation of the device (proposed by Paul Kocher in 1999)
- Why?
  - CMOS gates are the most popular building blocks of IC manufacturing
  - Power dissipation of CMOS gates depend on inputs





AISY framework

- a deep learning-based framework for profiling side-channel analysis
- brings state of-the-art deep learning-based side-channel attacks
- enables the users to run the analyses and report the results efficiently
- web application provides a user-friendly way to visualize analysis, plots, results, and tables

Leakage Models

Observable information that leaks via side-channels like power consumption

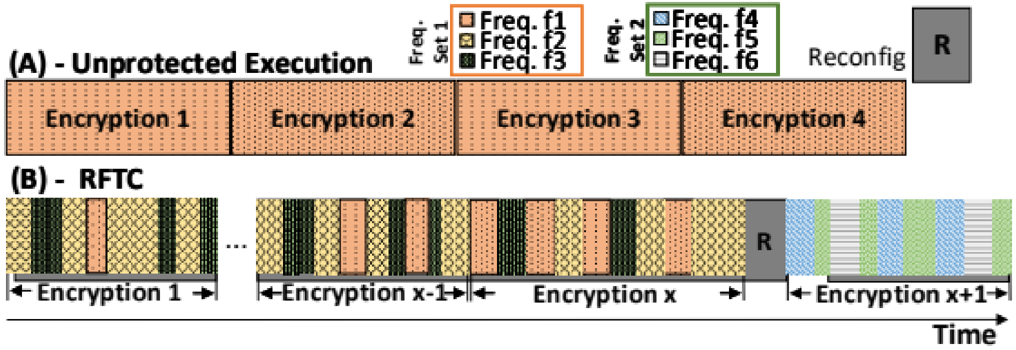
- Hamming Weight ("HW"):
  - number of '1' bits (set bits)
  - eg: "10101100" has a Hamming weight of 4
- Hamming Distance ("HD"):
  - differences between the two states
  - eg: State 1: "10101100" State 2: "10111100" → HD is 2 (bit 4 and bit 5).
- Bit:
  - each individual binary bits is treated as a separate class
  - eg: a bit value has 2 different classes that is 0 or 1

Countermeasures

- Masking : randomizing or masking the sensitive data during cryptographic operations
- Noise Injections: introduces additional noise in the side-channel signals
- Random Delay Insertion (RDI): inserts random delays into the execution of instructions.
- Random Clock Dummy Data (RCDD): inserts random dummy data into the clock signal.

What is RFTC?

- **R**andom **F**requency **T**uning **C**ountermeasure
- Introduces random frequency variations in the clock signal during the execution of cryptographic operations
- Instead of using a fixed clock frequency, RFTC dynamically changes the clock frequency at different phases of the operation.

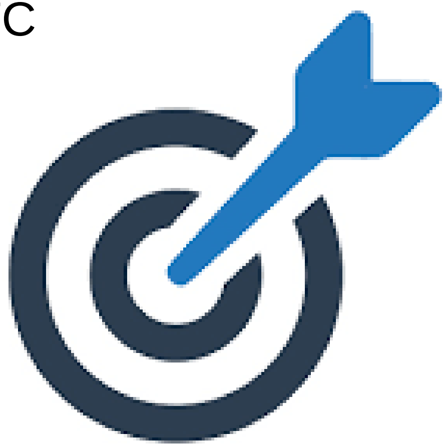


Why RFTC?

- None of the countermeasures were tested and proven to be secure against Correlation Power Analysis (CPA) based attacks (Preprocessed methodologies):
  - Dynamic Time Warping based CPA attacks (DTW-CPA)
  - Principal Component Analysis based CPA attacks (PCA-CPA)
  - Fast Fourier Transform based CPA attacks (FFT-CPA)
- RFTC is tested against all three attacks and shown to be secure for up to four million encryptions.
- But not tested against ML attacks

Project - Our Aim

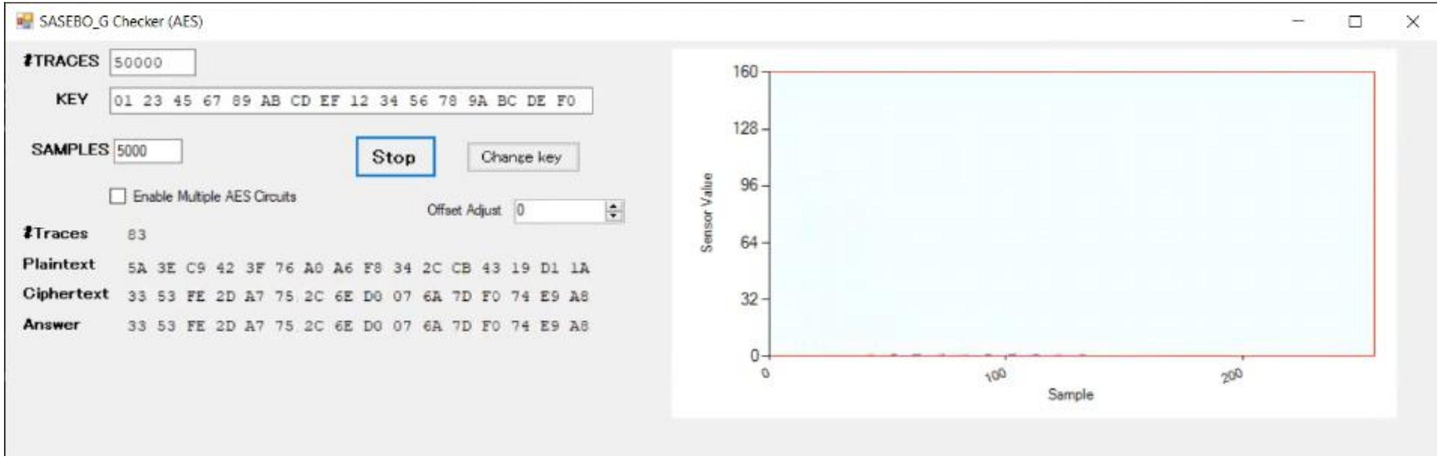
- Testing RFTC against Machine Learning models using AISY framework
- Improving MLP and CNN models to attack RFTC



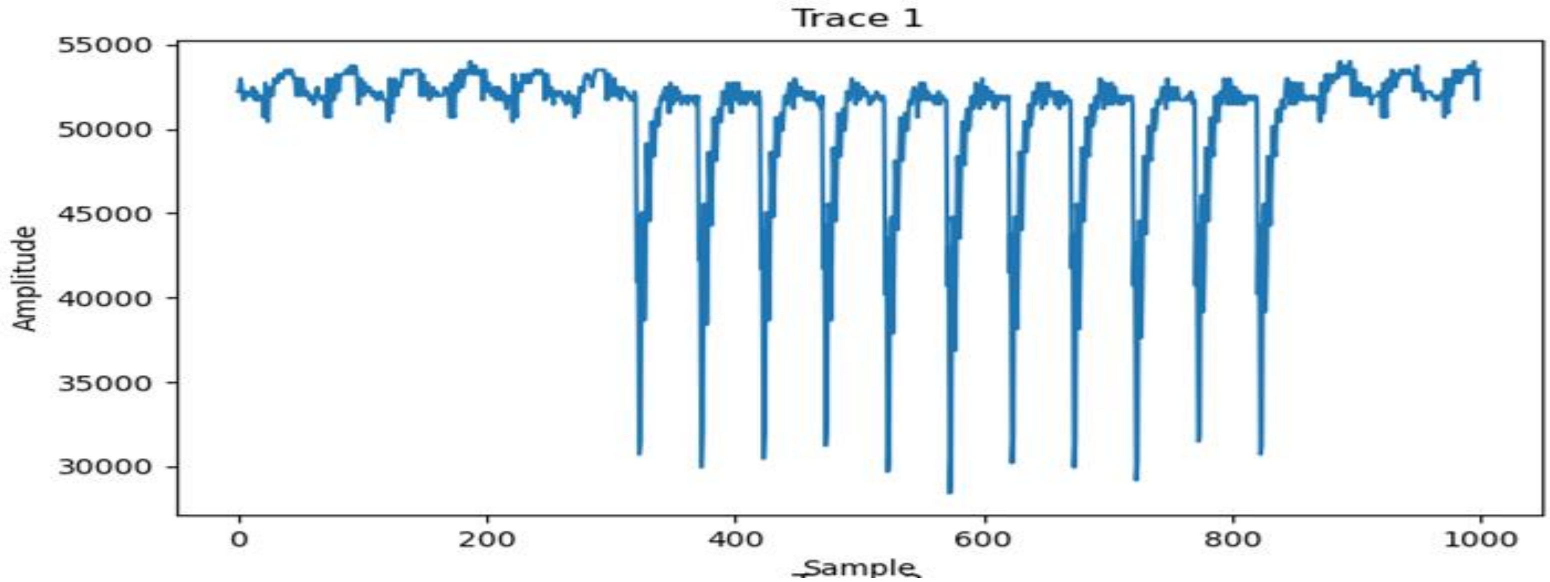
OUR PROGRESS

Taking Power Traces

- Have used a FPGA prototyping board with an isolated power line, signal amplifier.
- Have used a program which sends secret key, and random plain text and receive the cipher text.
- Have used another program to save plaintext, ciphertext, power traces and also key, because we want to verify if we received the key.
- Power traces are saved as a binary file.

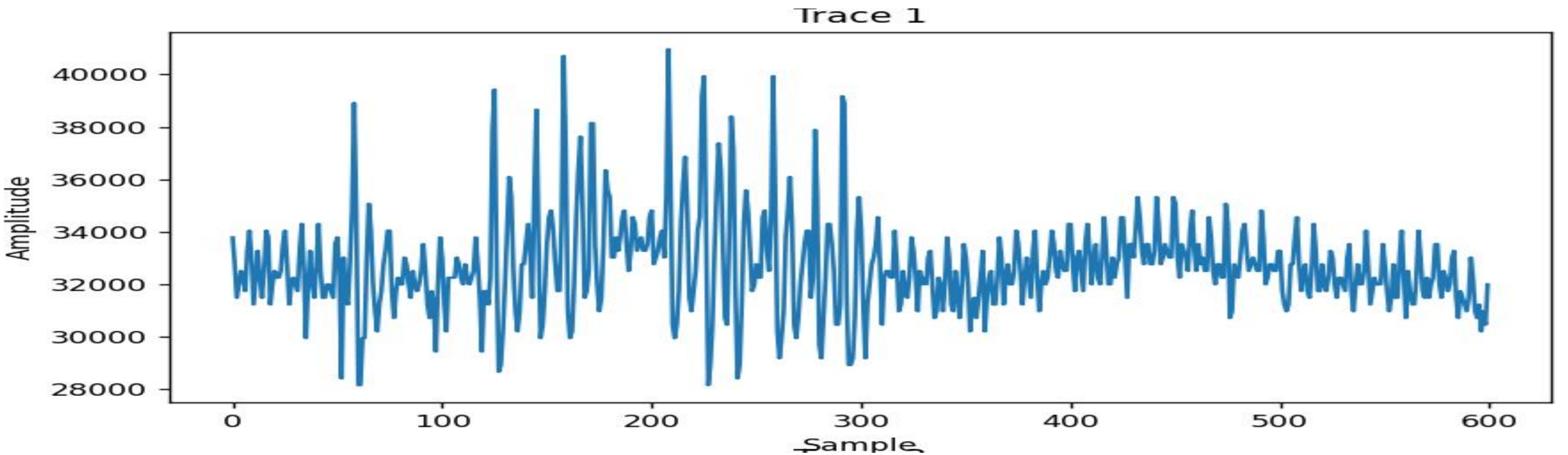


Unprotected Power traces



- Power traces of all 10 rounds are aligned

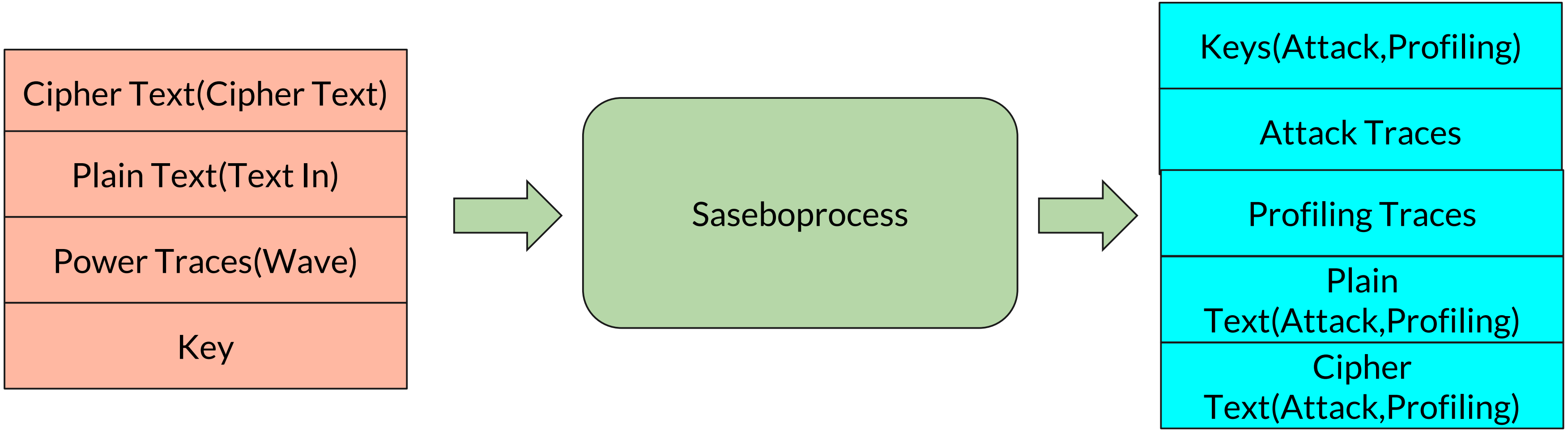
Protected Power traces (Using RFTC) - 1M traces



- Encryption happens during variable random frequency

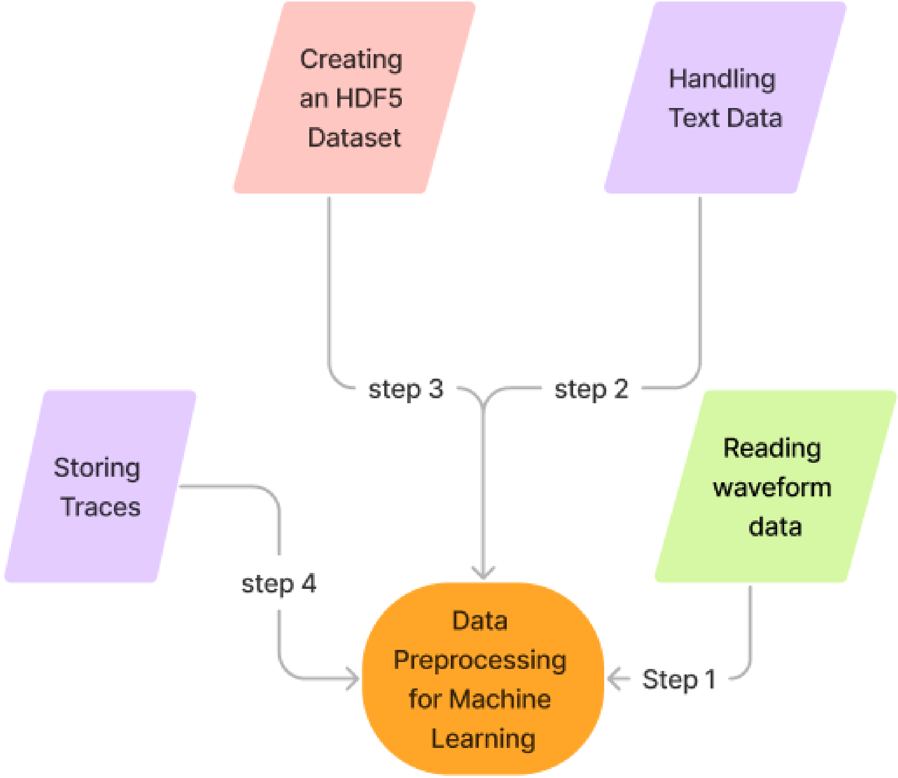
Converting Traces to h5

- Convert traces into h5 format because it is the supported format by aisy framework.
- In the saseboprocess power traces,cipher text,plain text and keys are separated into profiling and attack



Converting Traces to h5

- Data Preprocessing
- Reading waveform data
- Handling Text Data
- Creating an HDF5 Dataset
- Storing Traces in to h5 format



Converting traces to h5

- new_dataset.h5
  - Attack_traces
    - metadata
    - traces
  - Profiling_traces
    - metadata
    - traces

metadata at /Attack_traces/ [new_dataset.h5 in D:\CA\UOP\4th year\Sem7\CO421\datasets]

Table Import/Export Data

0-based

	0		
	plaintext	ciphertext	key
0	[242, 245, 114, 221, 191, 12...	[209, 204, 150, 81, 133, 140, 1, ...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
1	[61, 143, 88, 208, 5, 97, 188...	[222, 150, 19, 186, 34, 131, 41, ...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
2	[209, 36, 160, 54, 50, 136, 1...	[173, 127, 71, 172, 35, 80, 11, 2...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
3	[42, 124, 142, 164, 53, 201, ...	[45, 126, 243, 6, 71, 185, 142, 2...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
4	[149, 180, 211, 23, 9, 192, 1...	[0, 62, 115, 211, 197, 252, 128, ...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
5	[249, 150, 193, 101, 168, 24...	[145, 60, 17, 174, 35, 55, 148, 1...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
6	[177, 88, 125, 191, 65, 249, ...	[135, 118, 172, 76, 11, 193, 222...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]
7	[105, 180, 188, 59, 31, 91, 1...	[54, 112, 165, 210, 152, 115, 99...	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 240]

...contd

- new_dataset.h5
  - Attack_traces
    - metadata
    - traces
  - Profiling_traces
    - metadata
    - traces

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	52224.0	51968.0	52224.0	41728.0	44288.0	28928.0	31232.0	45312.0	44800.0	38144.0	42496.0	49152.0	46848.0
1	52736.0	52224.0	51968.0	41984.0	45056.0	30720.0	32768.0	46080.0	45568.0	38912.0	43264.0	49408.0	47104.0
2	52480.0	51712.0	52224.0	41728.0	44288.0	29696.0	32000.0	45824.0	45312.0	38400.0	42752.0	49152.0	46848.0
3	52480.0	51968.0	52480.0	41728.0	44544.0	29952.0	32512.0	45312.0	45568.0	38400.0	42496.0	49408.0	46592.0
4	51200.0	51456.0	51456.0	41984.0	43008.0	28672.0	31744.0	45312.0	43776.0	37120.0	42240.0	48640.0	45824.0
5	51712.0	51968.0	51712.0	42496.0	43264.0	28672.0	32000.0	45568.0	44288.0	37376.0	42496.0	49152.0	46336.0
6	52480.0	52480.0	52736.0	41728.0	45056.0	29696.0	31488.0	45568.0	45312.0	38144.0	42496.0	49408.0	47360.0
7	51456.0	51200.0	51712.0	40704.0	44544.0	29696.0	31744.0	45056.0	44544.0	38144.0	41984.0	48384.0	46592.0
8	52480.0	52736.0	52736.0	43520.0	43520.0	28672.0	33024.0	46592.0	44544.0	37888.0	43264.0	49408.0	46592.0
9	52224.0	52736.0	51968.0	42496.0	43520.0	28928.0	32512.0	46336.0	44800.0	38656.0	43008.0	49152.0	46848.0
10	52480.0	52480.0	52736.0	43520.0	45056.0	30976.0	33792.0	46848.0	46080.0	39424.0	43776.0	50176.0	47360.0
11	52736.0	52736.0	52480.0	41472.0	45056.0	31232.0	32768.0	46080.0	46336.0	39936.0	43520.0	49664.0	47616.0
12	51968.0	51712.0	51712.0	41216.0	44032.0	28928.0	31744.0	45056.0	44800.0	37632.0	42240.0	48640.0	46336.0
13	51712.0	51712.0	51712.0	41472.0	43776.0	28672.0	31744.0	45312.0	44800.0	37632.0	42240.0	48640.0	46080.0
14	51712.0	51968.0	51968.0	39680.0	44544.0	29952.0	30464.0	44544.0	45056.0	37888.0	41472.0	48640.0	46592.0
15	51968.0	51968.0	51968.0	41472.0	45056.0	31744.0	33024.0	46080.0	46080.0	39168.0	43008.0	49408.0	47104.0
16	51968.0	51968.0	51712.0	43264.0	43776.0	28928.0	33024.0	46080.0	44544.0	37888.0	43520.0	48896.0	46080.0
17	52224.0	52224.0	52224.0	40448.0	45568.0	32000.0	32512.0	45568.0	46336.0	39168.0	42496.0	49408.0	47616.0
18	52736.0	52736.0	52736.0	40192.0	45568.0	30720.0	31488.0	45568.0	46336.0	38656.0	42752.0	49920.0	47104.0

Attack unprotected AES using aisy framework

- AISY framework have already defined multilayer perceptron and convolution neural network models

```python
import sys
sys.path.append('D:/CA/UOP/4th year/Sem7/CO421/AISY_framework')


import aisy_sca
from app import *
from custom.custom_models.neural_networks import *

new_dataset_dict = {
    "filename": "new_dataset_2rounds.h5",
    "key": "000102030405060708090A0B0C0D0EF0",
    "first_sample": 0,
    "number_of_samples": 100,
    "number_of_profiling_traces": 60000,
    "number_of_attack_traces": 40000
}

aisy = aisy_sca.Aisy()
aisy.set_resources_root_folder(resources_root_folder)
aisy.set_database_root_folder(databases_root_folder)
aisy.set_datasets_root_folder(datasets_root_folder)
aisy.set_database_name("database_ascad.sqlite")
#aisy.set_dataset(datasets_dict["ascad-variable.h5"])
aisy.set_dataset(new_dataset_dict)
aisy.set_aes_leakage_model(leakage_model="ID", byte=4, round=1,
                           target_state="Sbox", direction="Encryption", cipher="AES128")

aisy.set_batch_size(400)
aisy.set_epochs(22)
aisy.set_neural_network(mlp)
aisy.run(key_rank_attack_traces=10000)
```

Attack unprotected AES using aisy framework

Attack using existing models (MLP)

- Can work on highly non-linear data.
- MLPs are capable of capturing these non-linear patterns, making them suitable for modeling such complex relationships
- Four hidden layers, each with 200 neurons, using SELU activation
- Use the Adam optimizer with a 0.001 learning rate and categorical cross-entropy as the loss function.

```
def mlp(classes, number_of_samples):
    model = Sequential(name="basic_mlp")
    model.add(Dense(200, activation='selu', input_shape=(number_of_samples,)))
    model.add(Dense(200, activation='selu'))
    model.add(Dense(200, activation='selu'))
    model.add(Dense(200, activation='selu'))
    model.add(Dense(classes, activation='softmax'))
    model.summary()
    optimizer = Adam(lr=0.0001)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model
```

Attack unprotected AES using aisy framework

Attack using existing models (CNN)

- More suitable for processing data that has a grid-like topology, such as an image.
- All the layers use the ReLU activation function, and the fully-connected layers have 128 neurons each.
- Use the Adam optimizer with a 0.001 learning rate and categorical cross-entropy as the loss function.

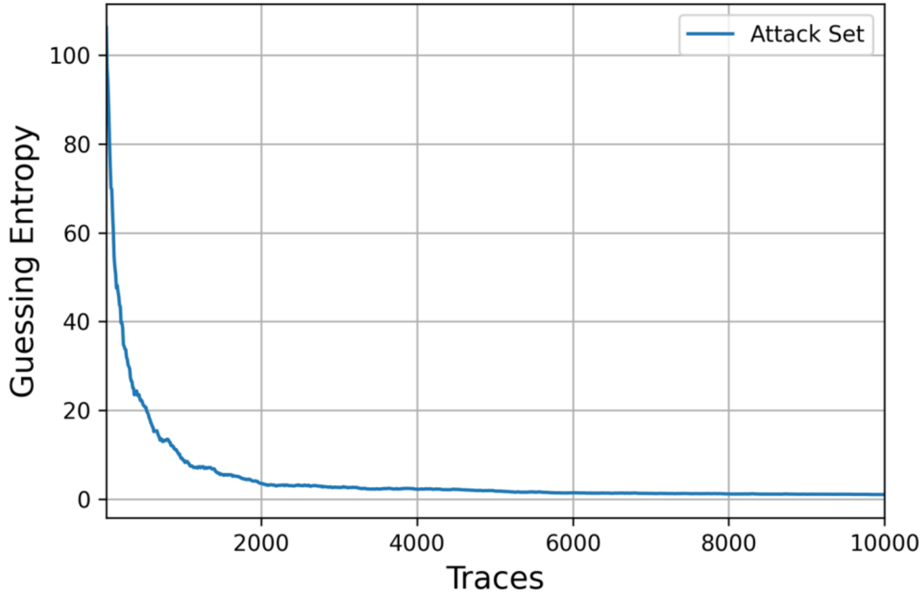
Increasing success rate for existing model

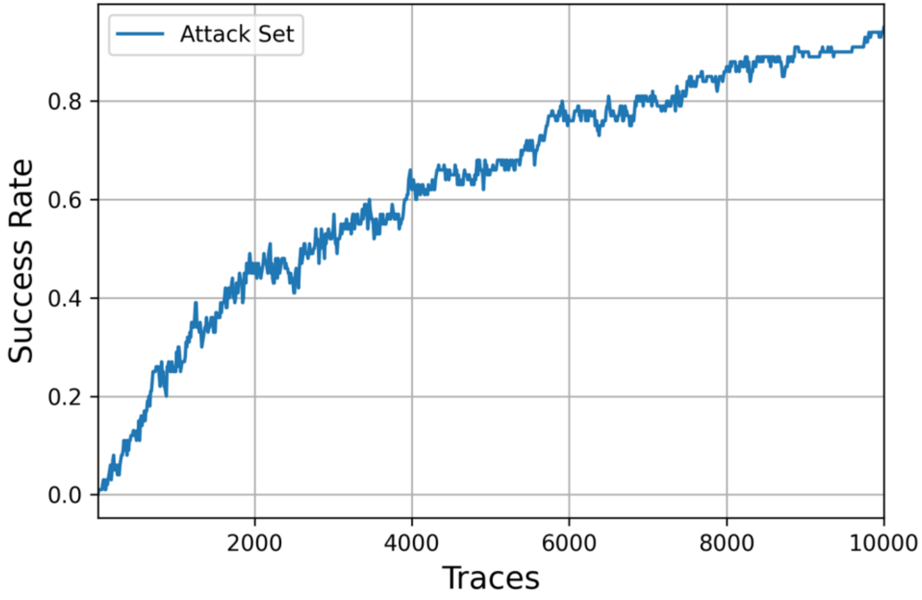
- Success rate - number of occurrences where we get the real key ranked first within the probability vector out of total number of attacks.
- Used;
  - existing MLP model in AISY framework.
  - traces consisting 1 round, 2 rounds, 3 rounds, and all the rounds of AES to attack.
- Attacked;
  - different key bytes.
  - different states of AES.
- Changed
  - leakage model used to attack.(HW, HD, ID)
  - the number of epochs.
  - number of key rank attack traces.

Increasing success rate for existing model

Analysis ID	Dataset	Datetime	Key Rank				Elapsed Time	NN Name	Results
			Key Byte	Metric	Guessing Entropy	Success Rate			
91	new_dataset_2rounds.h5	Sep 04, 2023 12:41:33	4	Attack Set	1	0.76	00:02:03	mlp	
92	new_dataset_2rounds.h5	Sep 04, 2023 12:47:33	4	Attack Set	2	0.41	00:02:05	mlp	
93	new_dataset_2rounds.h5	Sep 04, 2023 12:50:35	4	Attack Set	1	0.73	00:02:36	mlp	
94	new_dataset_2rounds.h5	Sep 04, 2023 12:54:26	4	Attack Set	1	0.95	00:01:59	mlp	
95	new_dataset_2rounds.h5	Sep 04, 2023 12:57:19	4	Attack Set	1	0.63	00:02:00	mlp	
96	new_dataset_2rounds.h5	Sep 04, 2023 13:00:17	4	Attack Set	1	0.8	00:01:58	mlp	
97	new_dataset_2rounds.h5	Sep 04, 2023 13:06:08	4	Attack Set	1	0.7	00:02:20	mlp	
98	new_dataset_2rounds.h5	Sep 04, 2023 13:09:56	4	Attack Set	1	0.65	00:02:40	mlp	

Increasing success rate for existing model

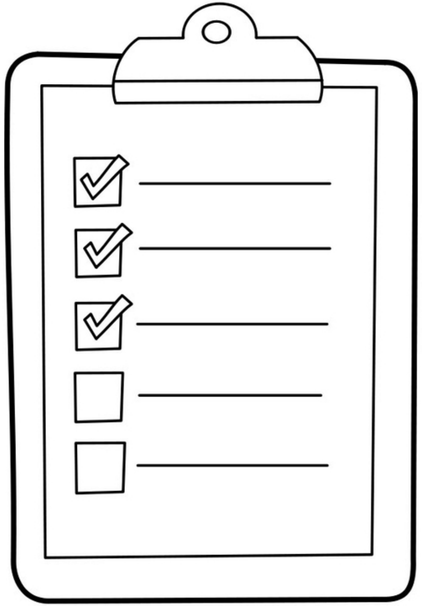




PLAN

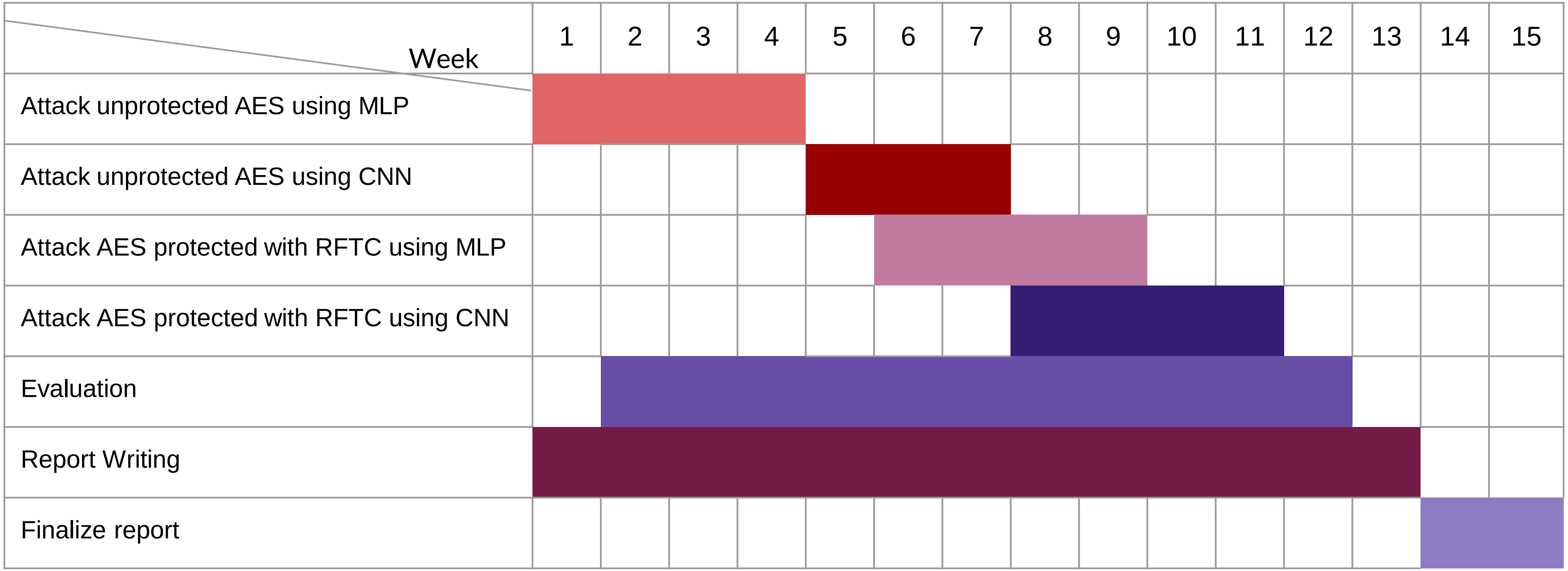
Remaining work

- Getting success rate = 1 for unprotected AES.
- Attack Protected AES using existing models.
- Attack protected AES using custom model.
- Preparing paper.



Work Plan

Semester 8



Thank You

Q & A