ML Based Side-Channel Analysis on AES Cryptosystem Through Power Consumption Data

Sourin Manna (Roll No:**CrS2218**)

Primary Supervisor:
Surya Prakash Mishra, DRDO, SAG Lab
Secondary Supervisor:
Debrup Chakraborty, ISI Kolkata

July 11, 2024

Table of Contents

1. Introduction
2. Data Understanding
3. Problem Statement
4. Power Base Attack
5. Study Objectives
6. Advanced Encryption Standard(AES-128)
7. Machine Learning Models
8. Study Methodology
9. Results and Discussion
10. Conclusion

Introduction

Introduction

- The Advanced Encryption Standard (AES) is a symmetric block cipher strong encryption method widely trusted for keeping data safe.
- Machine learning (ML) is a field of computer science that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention.
- This research explores the vulnerabilities of AES to side-channel attacks, specifically using power consumption data.
- The motivation behind the research, highlights the significance of securing cryptographic systems against side-channel attacks through the application of machine learning.

Data Understanding

Data Understanding

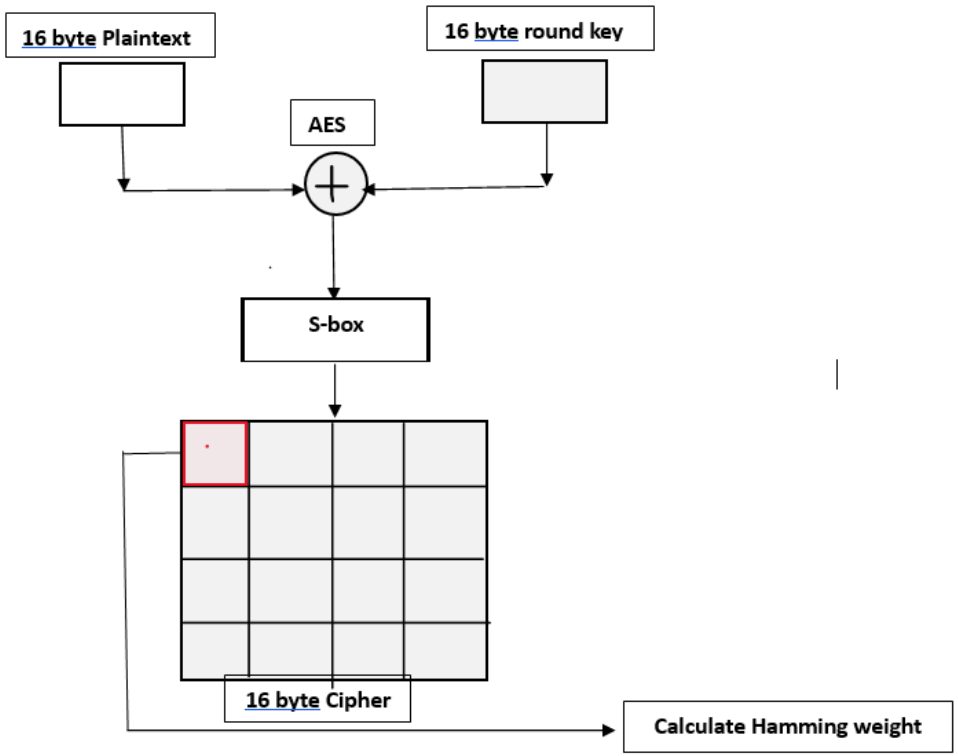

Figure: Calculating dependent Variable

Data Understanding

- **Hamming weight:** Hamming weight is the number of 1s in the binary representation of a number.
- **Independent Data:**The power consumption dataset contains 20,000 rows, representing the power consumption during the encryption of 20,000 plaintexts using AES, and 17,000 columns, signifying 17,000 features measuring power consumption every nanosecond.
- **Dependent Data:** We perform an XOR operation between the plaintext and the key to obtain a binary string of 0s and 1s. Next, we pass this binary string through the S-box. Finally, we calculate the Hamming weight of the first 8 bits of the resulting string,

Problem Statement

Problem Statement

- Applying Machine Learning algorithms to analyze the vulnerability of AES cryptosystem applied to power consumption data.
- **Aim:**
  - Our initial aim is to predict the Hamming weight of the first 8-bit ciphertext after the XOR operation between the plaintext and the key, followed by the S-box transformation.
  - Our final aim is to extract encryption key information from observed power patterns during encryption.
- Especially in this problem, we have 20,000 rows and 17,000 columns of independent data. The dependent data is the Hamming weight, Then we apply machine learning algorithm to predict Hamming weight.

Power Base Attack

Power Base Attack

- **Side-channel analysis** (SCA) attacks cryptographic algorithms by exploiting hardware vulnerabilities, such as device power consumption, electromagnetic radiation, timing, and sound, to reveal hidden information.
- Like other side-channel attack Timing Attacks,Cache Attacks,Fault Injection Attacks.Power Base Attacks be more effective, more practical to mount and cheaper to install than other side channels.
- Power Base Attacks exploit the power consumption patterns of cryptographic devices to try to extract secret keys to analyze power consumption by a machine.
- **The power consumed will be linearly dependent on the Hamming Weight of data being processed.**

Study Objectives

Study Objectives

- Identify potential weaknesses in AES cryptosystem.
- Develop and evaluate machine learning models to predict Hamming weight of ciphertext.
- Demonstrate feasibility of ML for side-channel analysis and highlight importance of countermeasures.

Advanced Encryption Standard(AES-128)

AES-128 bit

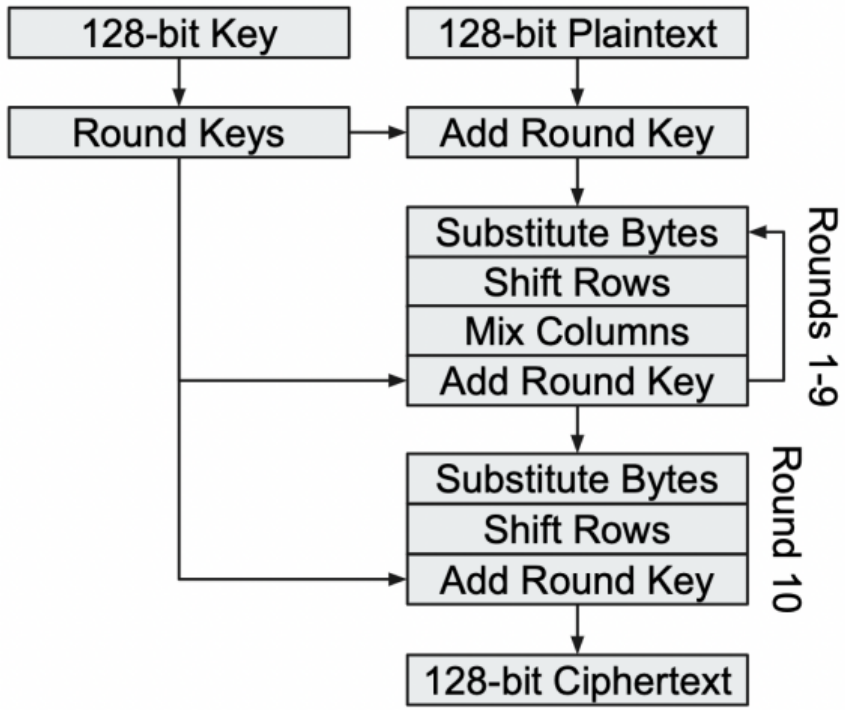

Figure: AES-128 bit

AES-128 bit

- AES, or the Advanced Encryption Standard is a symmetric encryption algorithm and a block cipher and is one of the most widely used symmetric-key algorithms today.
- The process AES operation first 'Key Expansion' and then 'Initial Round' we XOR with plain text with the first 128 bits of key expansion called AddRoundKey.
- The 'Main Rounds'(1 to 9) follows by SubBytes, ShiftRows, MixColumns, AddRoundKey.
- 'Final Round' (10th Round) SubBytes, ShiftRows, AddRoundKey (No MixColumns in the final round).

Machine Learning Models

Machine Learning Models

- Machine learning is an area of computer science a subset of Artificial Intelligence(AI) that uses algorithms to learn from data to build computer systems that can learn and improve on their own.
- Here we apply various ML algorithms:
  - **Logistic Regression:** Logistic Regression is a statistical model used for binary or multi-class classification.
  - **Random Forest:** Random Forest employs an ensemble of decision trees for classification and regression tasks.
  - **Naive Bayes:** Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence among features.
  - **Artificial Neural Network(ANN):** Artificial Neural Network (ANN) mimics the human brain's neural network structure to learn complex patterns.
  - **K-nearest neighbor:** K-nearest neighbor is a simple, non-parametric algorithm that classifies a data point based on the majority class of its K nearest neighbors in the feature space.
  - **XGBoost:** XGBoost employs gradient boosting to iteratively build a series of decision trees for predictive modeling.

Study Methodology

Experimental Setup

- We analyze physical leakages during cryptographic operations using ML model.
- **Dataset Description(dependent & independent data):**
  - The dataset has 20,000 rows and 17,000 columns. Each row shows power consumption during AES encryption, with the 17,000 columns representing power measurements taken at nanosecond intervals.
  - The target variable is the Hamming weight of the first 8 bits of the ciphertext, calculated by XORing the plaintext with the AES key twice and then computing the Hamming weight of the result.
- **Data Preprocessing(Normalize & Merge):**
  - First we normalize power consumption values for data security, then merge independent (power data) and dependent (Hamming weight) tables by plaintext, and then split the dataset into training and testing parts.

Evaluation Process

- **Cryptographic Operation Extraction:**To get the dependent variable, XOR the plaintext byte with the key byte, pass the result through the AES S-box, and calculate the Hamming weight of the first 8 bits.
- **Model Selection:**We apply several multi-class machine learning models to predict the Hamming weight. The chosen models include:
  - Logistic Regression, XGBoost, Random Forests, Neural Networks.
- **Training and Testing:**
  - We are given a table with 20,000 rows and 17,001 columns. We will now apply various machine learning algorithms and determine the accuracy of each model.
  - **Data Split:** The dataset is split into training (80%) and testing (20%). This split ensures that the models are tested on unseen data.
  - **Model Training:** Each model is trained on the training set. Cross-validation helps in validating the model's performance across different subsets of the data.
  - **Model Testing:** The models are evaluated on the testing set to assess their ability to predict the Hamming weight on new data.

Performance Metrics

- Basically we evaluate our all the models using the following metrics:
- **Accuracy:**Accuracy measures the proportion of correct predictions
- **Precision:** Precision measures the proportion of true positive predictions out of all positive predictions.
- **Recall:** Recall measures the proportion of true positive predictions out of all actual positive cases.

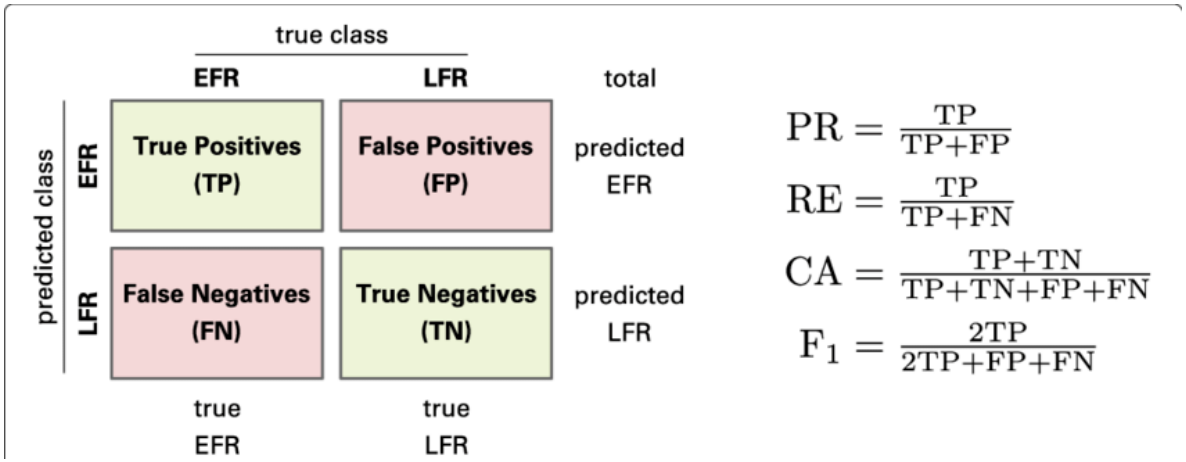

Figure: Accuracy, Precision & Recall

Results and Discussion

Results

- Among the models tested, XgBoost emerged as the most effective, achieving a significant accuracy rate of 63.9% in predicting the hamming weight.
- This result outperformed Logistic Regression (44.2%), Random Forest (47.9%), and Artificial Neural Networks (After 5 epoch) (43.95%).
- Other Machine Learning algorithm which performs very less Naive Bayes classifier(29.6%) and K-Nearest Neighbors(32.6%).
- The table below summarizes the performance of each machine learning model:

Results

```
import xgboost as xgb
xgb_model=xgb.XGBClassifier(random_state=0)
xgb_model.fit(x_train,y_train)
```

XGBClassifier

```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              objective='multi:softprob', predictor=None, ...)
```

```
# Predicting on the testing set
y_pred = xgb_model.predict(x_test)

# Calculating accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.6391666666666667
```

Results

Model	Accuracy (%)	Precision(%)	Recall(%)
Logistic Regression	44.2	0 or 44.1*	0 or 100*
Random Forest	47.9	53.4	76
Naive Bayes	29.6	67.8	33.7
Artificial Neural Networks	43.95	-	-
K-nearest Neighbour	32.9	-	-
XgBoost	63.9	-	-

Table: Performance metrics of machine learning models in predicting the hamming weight of the first 8 bits of the ciphertext.

* Precision & Recall are two different vale for different class.

Results

Accuracy: 0.44233333333333336

Classification Report:

	precision	recall
2	0.00	0.00
3	0.00	0.00
4	0.44	1.00
5	0.00	0.00

Figure: Classification Report of Logistic Regression model of four classes 2, 3, 4, and 5

Conclusion

Conclusion

- Conclusion 1: Our findings indicate that **machine learning** models, particularly XgBoost, have **significant potential** in identifying vulnerabilities in AES encryption systems through side-channel attacks.
- Conclusion 2: XgBoost achieved the **highest accuracy rate** of 63.9% in predicting the Hamming weight of the first 8 bits of ciphertext after XOR with the key and S-box transformation.
- Conclusion 3: Our project evaluated the machine learning-based side-channel attack resistance of the AES-128 cryptosystem using power consumption data, revealing that the **AES-128** implementation is **vulnerable** to such attacks.
- Conclusion 4: We can predict the AES key by using a **128-class machine learning** model, with Hamming weight as input and the key as output.

Further Works

- Our future research aims to extend from analyzing the 1st round AES-128 to exploring vulnerabilities across **multiple rounds** of AES-128 encryption.
- In addition to power consumption data, we intend to analyze **other side-channel information** such as Timing Attacks, Cache Attacks, and Fault Injection Attacks.
- We propose to leverage **advanced machine learning** and deep learning algorithms, including Generative Adversarial Networks (GANs), to enhance prediction accuracy in side-channel attacks.

Thank you!