CS4150: Machine Learning for Side-Channel Attacks

Bas van't Spijker
1497944

Yuhang Tian
5219728

May 27, 2021

1 Model Description

- Leakage Model: There are two kinds of models in this experiment. The one is the intermediate leakage model - the output values of S-Boxes while another is Hamming weight leakage model - the Hamming distance of S-Boxes' outputs. Intuitively, the intermediate leakage model, when the training samples are sufficient, performs better than the Hamming weight one, because it categorizes the outputs more precisely and the labeling range is from 0 to 255 whereas the Hamming weight leakage model only contains 9 labels. However, when the number of training traces is limited, the model which has more classes performs worse since some categories may not get enough data to support training.
- Feature Selection Algorithm: In this experiment, chi-square (X^2) statistic will be used for the feature selection. We tried to use PCA instead, but its performance is worse than chi-square, so we switched back to chi-square. Before sifting the traces, we used *MinMaxScaler()* to re-scale the dimensions for all training and testing data so as to obtain the top significant features more fairly.
- Probabilistic Model: The algorithm to generate the probabilistic model in this experiment is Random Forest which works well for predicting categorical features and labels and runs faster than SVM. Its parameters are set as n_estimators=[50, 100, 150] and max_depth=[5, 10, 20]. Cross-validation will be applied to figure out the best parameters combination for each subkey which cv is 3 and evaluation metric is accuracy.
- Training & Testing Setting: The number of training traces and testing traces will be 5000 and 10, respectively. 50 points of interest will be selected out from the 5000 sampling points.
- Visualization Method: It will show the guessing entropy results vs. the number of testing traces.
- Desynchronization: The level of desynchronization is 10. For each trace, it will randomly pick a value from [0, 10) and shift all the sampling points with that value.

2 Result Visualization

Figure. 1 shows the result of the Random Forest probabilistic model based on the intermediate leakage model predicting each subkey when observes a different number of testing traces. In general, it only uses 1 trace to obtain the correct subkeys.

Figure. 2 shows the result of the Random Forest probabilistic model based on the hamming weight leakage model predicting each subkey when observes a different number of testing traces. In general, it uses more traces than the intermediate leakage model to obtain the correct subkeys, especially for obtaining key[4] for which it uses 10 traces.

In fig. 3, it shows the result of the Random Forest probabilistic model based on the intermediate leakage model predicting each subkey when observes a different number of testing traces. However, desynchronization is applied to all traces. Comparing to non-desynchronization traces, the model currently requires more testing traces to find out the correct subkeys, but fortunately, some of them can still be found.

In fig. 4, it shows the result of the Random Forest probabilistic model based on the hamming weight leakage model predicting each subkey when observes a different number of testing traces. However, desynchronization is applied to all traces this time. Comparing to non-desynchronization traces, it cannot figure out the correct subkeys anymore.

3 Conclusion

- For equal training and testing samples, the intermediate leakage model spends more time for training and validating than Hamming weight leakage model.
- The intermediate leakage model, when the training traces are ample, performs much better than Hamming weight leakage model. It uses fewer testing traces to figure out the whole key and sometimes even uses a single trace.

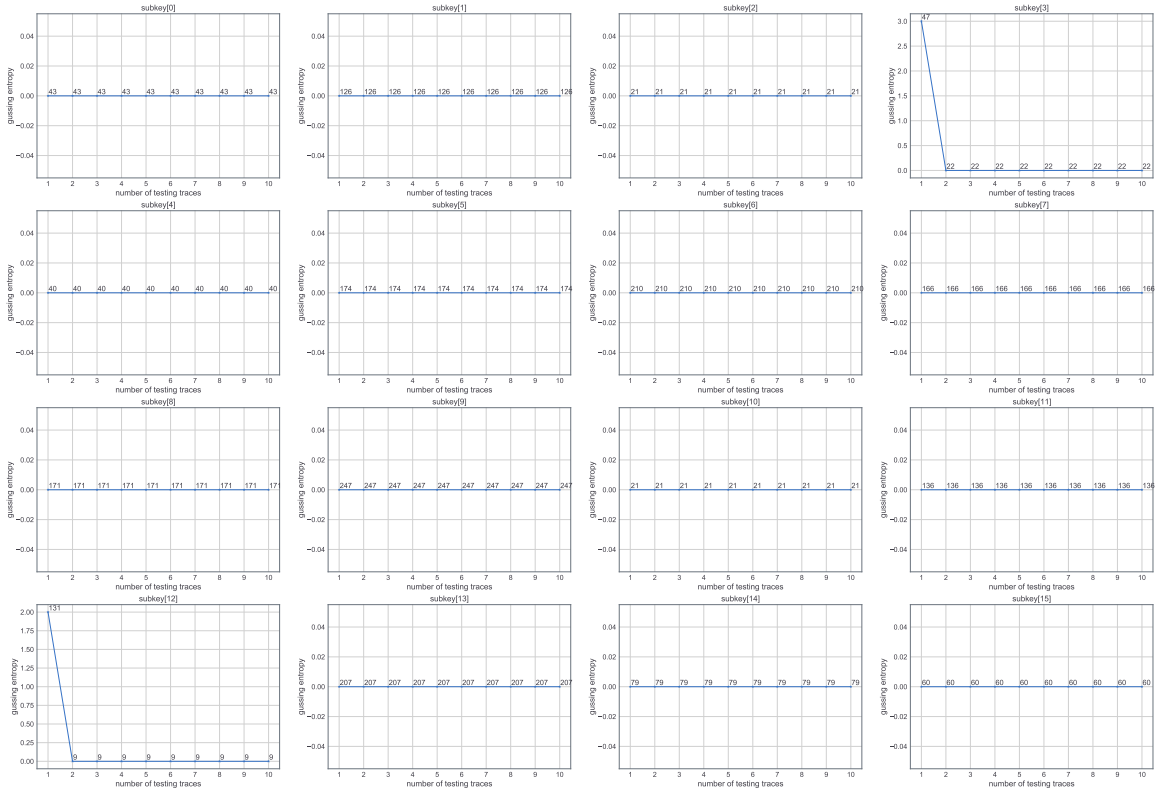

Figure 1: Intermediate Leakage Model

- Adding level 10 desynchronization makes Hamming weight leakage model useless, whereas the intermediate leakage model has the ability to resist desynchronization. The intermediate leakage model can still reduce the guessing entropy when the number of testing traces increases.

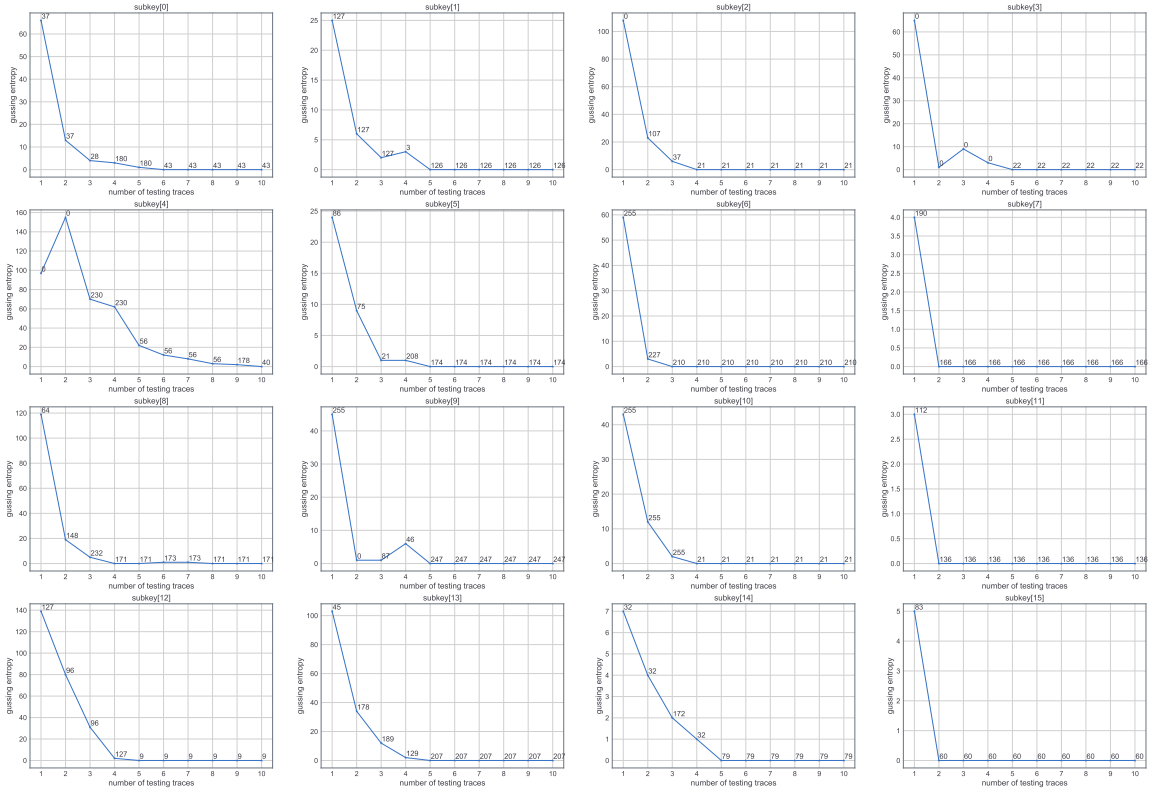

Figure 2: Hamming Weight Leakage Model

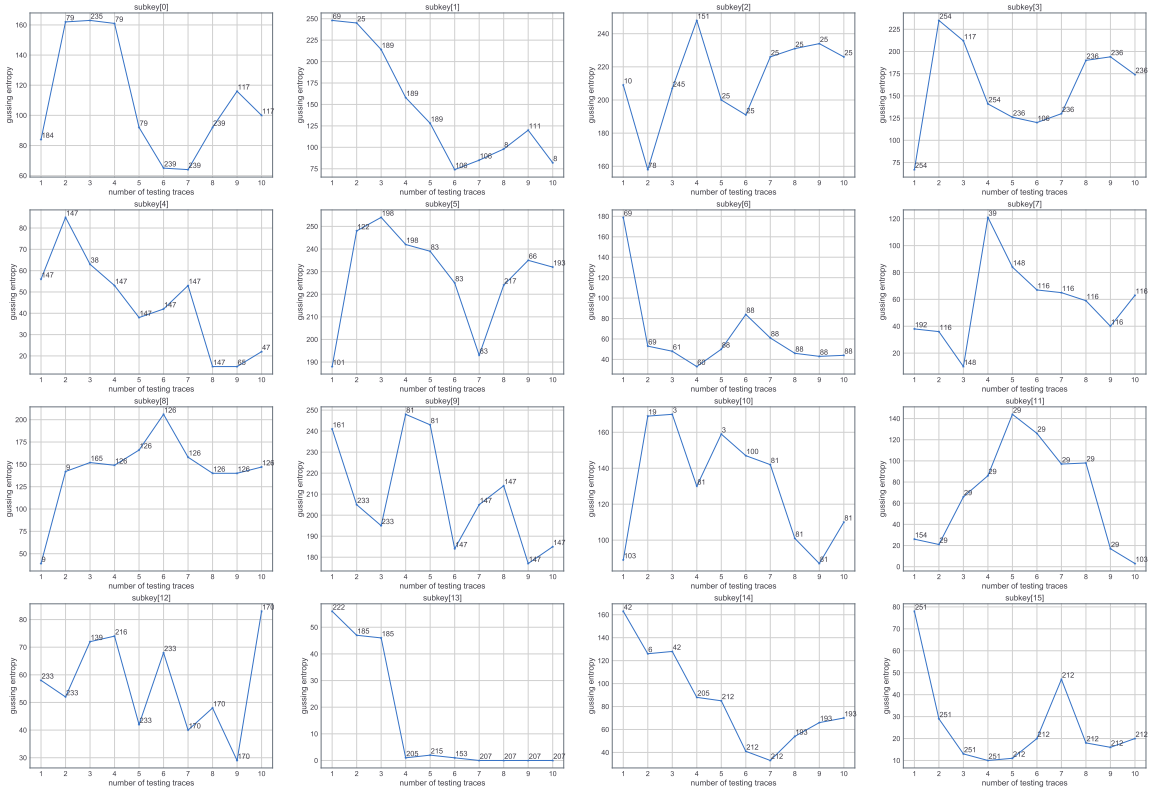

Figure 3: Intermediate Leakage Model with Desynchronized Traces

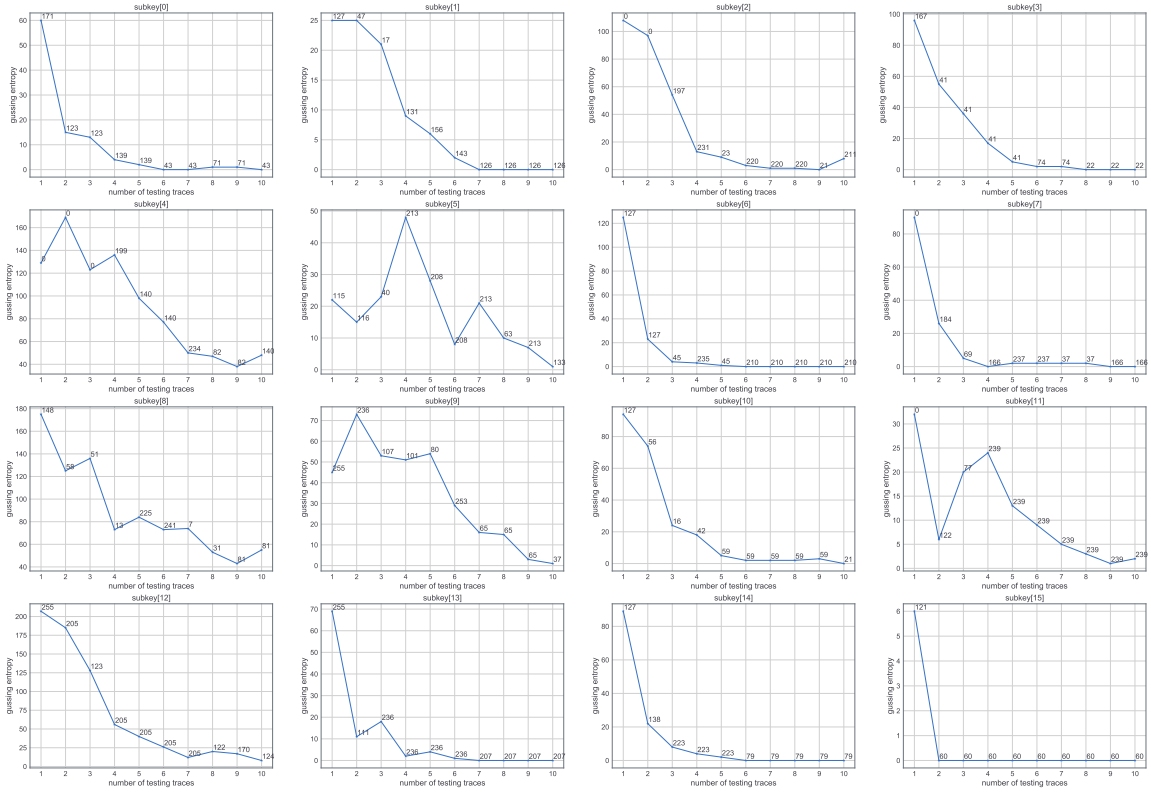

Figure 4: Hamming Weight Leakage Model with Desynchronized Traces