Distinguishing Attacks on AES algorithms using Deep Learning

Kimberly Devi Milner
Department of Computer Science
New York University
km1851@nyu.edu

***Abstract*—In this paper, we investigate using deep learning to classify modern encryption algorithms. We create a neural network model trained on ciphertext. We hypothesize that given the achievements in deep learning technology we will have high accuracy in our classifier distinguishing the AES mode of operation used to encrypt the ciphertext. We conclude that our classifier needs further feature tuning before a determination can be made.**

Keywords—AES operation modes (CBC, CFB, OFB), encryption algorithm classification, neural networks, Keras, pycryptdome

I. Introduction

In cyber warfare cryptanalysis enables the investigation of ciphertexts and ciphers of obtained systems. The deciphering of ciphertext without the cipher or key is a formidable exercise. A simpler exercise is to merely distinguish the encryption algorithm used. Since 2007 researchers in the US, UK and China have experimented with the success of using Machine Learning to do this. With varying feature sets, both in training on the key or not, and the encryption algorithms used, researchers have claimed varied success. Chandra et al [4] for instance claims cryptanalysis powered by machine learning can be used for distinguishing attacks. They did not seem to address AES, however. In 2012 [1] Chou et al argued that the success of distinguishing attacks should be claimed only if harder to break algorithms (like AES' CBC operation mode) were included. Since then, other researchers have also been able to successfully perform distinguishing attacks, including on AES, see [2], using the Support Vector Machine (SVM) techniques of Machine Learning.

The AES algorithm has developed modes of operation that adapt the algorithm to be better suited to varied type of data [5]. With the recorded success of using Machine Learning to distinguish encryption algorithms, we wanted to see if it was possible to use the powerful tooling of Neural Networks to distinguish modes of operation of AES encrypted ciphertexts.

II. Related Work

Chandra et al [4] generated ciphertext with the Enhanced RC6 block cipher and Stream Cipher (SEAL), using the "summation" of ASCII blocks of the ciphertext to train both a Cascade Correlation Neural Network Model, and a Gradient Descent back propagation model. Using two hidden layers, Chandra et al were able to achieve distinguishing accuracies as high as 93 percent when using the same plaintext for both encryption schemes over 1000 epochs, and at worst 73% when encrypting different plaintexts.

Chou et al [1] used more diverse plaintext, including images and audio, and attempted to distinguish between the AES modes of operation, Cipher-Block Chaining CBC, and Electronic Codebook (ECB), as well as between AES and DES. Chou et al did not find that current machine learning effective to perform distinguishing attacks and noted that previous experiments did not "consider CBC mode of operation with random IV, which is the recommended configuration capable of providing the basic level of security." [1]

Eight years later Tan et al [3] were able to classify against more algorithms, but still did not test against CBC operation for AES. Their SVM classification system tested ciphertexts created through AES (in ECB mode), Blowfish, Triple-DES, RC5, and DES, for both the same key and different keys. They found that AES was more easily recognized than the others, and the classifier performed well when trained on datasets produced with the same key.

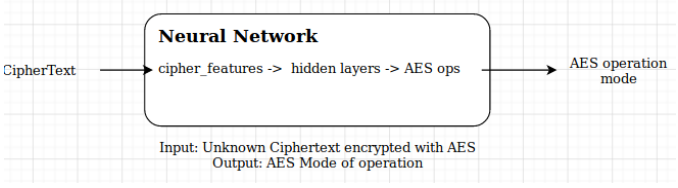
Understanding the importance of testing for AES's stream cipher mode of operation, Zhou et al built a classifier that also classified the stream cipher Grain128 against other stream and block ciphers, concluding that with high accuracy Grain-128 could be identified, thereby indicating it exhibited "more unrandomness" than the other ciphers. [2]

It is likely that other work we did not uncover exists in the field of encryption classification. We did not see work that attempted to distinguish modes of operation in AES using neural networks, and it is here that we hope to offer contributing research.

III. Hypothesis

Given the advances made in machine learning since [1] Chou et al performed their experiment, we expect a high accuracy in being able to distinguish AES ciphertext encrypted with different modes of operations.

Figure 1: Classifier to create



IV. Methods

A. Data Learning Technology

Deep learning and data generation for our classifier was done in python Jupyter notebooks using the deep learning library Keras, and the crypto library pycryptodome [6].

The pandas library was used for data framing, while the sklearn was used for k-fold cross validation.

B. Data Generation

A simple string message was chosen as plaintext. Every record in the generated dataset of ciphertext is encrypted in one of AES' three modes of operation: Cipher Block Chaining (CBC), Cipher Feedback (CFB) or Output Feedback (OFB). Every record is created with the same key and message, but with a random initialization vector (IV).

The ciphertext was encoded in hex, then turned into a decimal. The resulting decimal is treated as a string and is broken into five blocks of eight characters, with each block serving as a feature. 30,000 such record entries were generated.

```
In [750]: df.head(4)
```

Out[750]:

	block1	block2	block3	block4	block5	label
0	22197278	88884226	71569029	19161026	472704	CBC
1	21669637	2008221	36307018	27065564	786571	CBC
2	28845731	98121802	46120501	49811933	782084	CBC
3	10788982	30331979	97778935	78639215	4463580	CBC

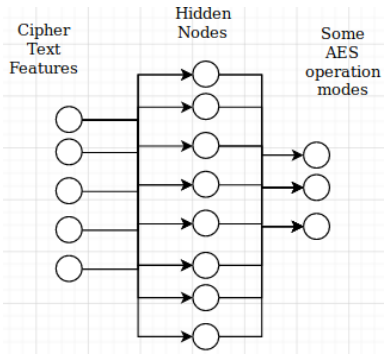
Figure 2: Sample traning data

C. Modeling

Our neural network has one hidden layer containing eight neurons. There are 5 inputs (as a ciphertext is categorized by five decimals blocks) and 3 output nodes since a classification of 'CBC', 'CFB' and 'OFB'.

The activation for the hidden layer is the standard rectifier activation function, while activation for the output layer is the softmax activation function (as opposed to sigmoid) since we have a multi-class classification problem.

Figure 3: Neural Network model used



Preliminary results and conclusion

The batch size before updating the model was 5 samples, and 200 epochs (or passes through the training data set) were performed. We evaluated our model with a 10 folds cross validation. Our accuracy was not what we expected. We recorded an accuracy of 37% with a relative standard deviation of 2.42%.

```
In [698]: # epoch is one forward pass and one backward pass of the training exampels
          # batch size is the number of training examples

          estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)
          kfold = KFold(n_splits=10, shuffle=True)
          results = cross_val_score(estimator, X, dummy_y, cv=kfold)
          print("Accuracy: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))

          Accuracy: 37.00% (2.42%)
```

Figure 4: Accuracy and relative std deviation

There are many potential reasons for the underperformance of the classifier. Indeed, Chandra et al [1] also noticed poor performance in encryption classification at first, which prompted them to better tune their features. Here, feature tuning – such as representing the cipher text blocks as ASCII or base64 instead of as decimals, and having varied messages of longer length (instead of just encoding a simple string) -- could easily affect performance.

Without the aforementioned it is not possible for us at this time to comment on whether neural networks can potentially be used to identify ciphertext encoded in varied AES modes of operation.

References

[1] Chou, J.-W., Lin, S.-D., & Cheng, C.-M. (2012). On the effectiveness of using state-of-the-art machine learning techniques to launch cryptographic distinguishing attacks. *Proceedings of the 5th ACM Workshop on Security and Artificial Intelligence - AISec 12*. doi: 10.1145/2381896.2381912

[2] Zhao, Z., Zhao, Y., & Liu, F. (2018). Research on Grain-128s cryptosystem recognition. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). doi: 10.1109/iaeac.2018.8577796

[3] Tan, C., & Ji, Q. (2016). An approach to identifying cryptographic algorithm from ciphertext. 2016 8th IEEE International Conference on Communication Software and Networks (ICCSN). doi: 10.1109/iccsn.2016.7586649

[4] B. Chandra and P. P. Varghese, "Applications of cascade correlation neural networks for cipher system identification," International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 1, no. 2, pp. 369 – 372, 2007

[5] Aes Modes of Operation. (2019). International Journal of Innovative Technology and Exploring Engineering, 8(9), 2213–2217. https://doi.org/10.35940/ijitee.i8127.078919

[6] "Classic Modes of Operation for Symmetric Block Ciphers." Classic Modes of Operation for Symmetric Block Ciphers - PyCryptodome 3.9.4 Documentation, https://pycryptodome.readthedocs.io/en/latest/src/cipher/classic.html.