On the Effectiveness of using State-of-the-art Machine Learning Techniques to Launch Cryptographic Distinguishing Attacks

Jung-Wei Chou
Department of Computer Science
National Taiwan University
Taipei 106, Taiwan
r99922018@csie.ntu.edu.tw

Shou-De Lin
Department of Computer Science
National Taiwan University
Taipei 106, Taiwan
sdlin@csie.ntu.edu.tw

Chen-Mou Cheng
Department of Electrical Engineering
National Taiwan University
Taipei 106, Taiwan
ccheng@cc.ee.ntu.edu.tw

ABSTRACT

Cryptographic distinguishing attacks, in which the attacker is able to extract enough "information" from an encrypted message to distinguish it from a piece of random data, allow for powerful cryptanalysis both in theory and in practice. In this paper, we report our experience of applying state-of-the-art machine learning techniques to launch cryptographic distinguishing attacks on several public datasets. We try several kinds of existing and new features on these datasets and find that the ciphers' "modes of operation" dominate the performance of classification tasks. When CBC mode is used with a random initial vector for each plaintext, the performance is extremely bad, while the performance for certain datasets is relatively good when ECB mode is used. We conclude that, in contrary to the findings of several existing works, the state-of-the-art machine learning techniques *cannot* extract useful information from ciphertexts produced by modern ciphers operating in a reasonably secure mode such as CBC, let alone distinguish them from random data.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]; I.2.1 [**Applications and Expert Systems**]; I.5.4 [**Applications**]; K.4.1 [**Public Policy Issues**]: Abuse and crime involving computers

Keywords

Computer Forensics, Cryptographic Distinguishing Attacks, Identification of Encryption Algorithm, Machine Learning

1. INTRODUCTION

In cryptography, if an attacker can extract enough information from a ciphertext and distinguish it from random data, then we say that he or she succeeds in launching a distinguishing attack. Such an attack might seem innocuous at a first glance, but it can actually lead to several powerful cryptanalytic attacks.

AISec'12, October 19, 2012, Raleigh, North Carolina, USA.

For example, Martin and Shamir gave a classical example of such amplification [1]. More recently, Albrecht, Paterson, and Watson gave another example in which they succeeded in attacking one of the most widely used Internet security softwares, the OpenSSH, by turning distinguishing attacks into plaintext-recovery attacks [2]. Therefore, distinguishing attacks have been playing an important role in modeling cryptographic ciphers, and many cryptographers believe that it is computationally infeasible to launch distinguishing attacks against reasonably secure ciphers such as DES and AES.

In this paper, we focus on an important, albeit slightly easier task in cryptanalysis: Identification of encryption algorithm. It is easier in the sense that we don't need to get too involved in what random data is from a technical or philosophical viewpoint. Furthermore, such a task can be important in scenarios like digital forensics because only the evidence from computer media is available. In these cases, we don't even know which cipher was used to encrypt the messages, whereas in textbook cryptanalysis scenarios, the encryption algorithm is always given. In order to recover useful information without using any meta-data, the technique of identification of encryption methods is needed. Overall, this problem has not been investigated much in the literature. Furthermore, the few papers that have paid some attention to it almost all use a set of similar features and claim some success for ciphers operating in simple modes. In this paper, we compare the performance of existing features in different scenarios and show that the classification accuracy can significantly differ when different modes of cipher operation are used. Without loss of generality, we only consider binary-class cases, as multi-class tasks can be easily done by extending the approaches used in binary cases.

We design different scenarios by introducing different modes of operations in encryption process. The mode of operation is a procedure that repeatedly uses a block cipher with a fixed key to encrypt a message whose length is larger than one block. The simplest one is electronic codebook (ECB) mode. In ECB mode, a message is divided into several blocks, and each block is encrypted independently. The advantage is speed because encryption of different blocks can happen in parallel. However, such a mode doesn't provide semantic security, as the same plaintext block always encrypts to the same ciphertext block. The cipher-block chaining (CBC) mode is the most commonly used one. In CBC mode, the message is also divided into blocks, but before each block is encrypted, the plaintext is XORed with the ciphertext of previous block. For the first block, an initialization vector (IV) is used to be XORed with the plaintext. Thus each ciphertext depends on all blocks processed up to the current block.

2. RELATED WORK

Genetic algorithm based methods are widely used in recovering secret keys in encryption algorithms, such as for substitution cipher [1], transposition cipher [4], knapsack cipher [5], and Feistel cipher [6], by localized searching in the key space. Neural networks are also used to break cryptosystems [7][8]. As will be detailed below, there are already some existing works on cipher classification based on statistics techniques..

There are some works done by Pooja on the classification of classical ciphers [9]. It includes substitution cipher, permutation cipher, polyalphabetic cipher, and a combination of permutation and substitution cipher. Several cost functions are proposed to distinguish classical ciphers by sorted or unsorted frequency of letters. An expected frequency of letters is also required, which is drawn from common English texts.

Some early work of classifying modern ciphers has done by Chandra [10] by combining several decision logics to classify modern ciphers. Dileep [11] proposed to use support vector machine (SVM) and bag-of-words model for identification of block ciphers, which builds common or class-specific dictionary of (1) fixed length words and (2) variable-length words. Saxena proposed to use linear programming on the segments of ciphertexts to generate many test vectors [12] and use SVM to find good test vectors. Sharif used a number of classifiers on 8-bit histogram features for identification of encryption methods and reported that random forests outperform all other classifiers **錯誤! 找不到參照來源。**. Manjula proposed to use several features such as entropy, correlation coefficient of uppercase letters, and size of files to identify encryption algorithms by decision tree [14].

As we will demonstrate in the rest of this paper, almost all these related works won't work against a reasonably secure cipher operating in CBC mode. We will also give the reasons why they seemingly work in their reports and suggest what we should do in the future when doing research in this direction.

3. EXPERIMENT

In this section, we focus on using existing bag-of-words model of feature and the common classification framework in [11]**錯誤! 找不到參照來源。** to solve the problem of cipher identification. The framework is shown in Figure 1.

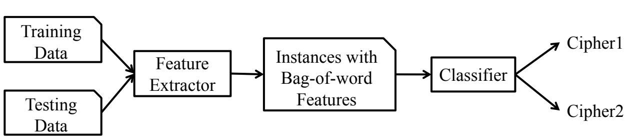

Figure 1. The Framework of Bag-of-word Approach

3.1 Environment and Setup

We use 3 kinds of dataset for validation, including text, images, and audio files. The Reuters21578 dataset [15] is a collection of news on Reuters newswire in 1987. In preprocessing step, we filter out documents smaller than 128 bytes in size. For images, we use Caltech 101 dataset [16], which is widely used in computer vision with 101 categories and total 19043 images in JPEG format. We note that the category information is not used in the experiments below. For audio files, MajorMinor dataset is used [17], which contains 2174 audio files in WAVE format.

The experiments are divided into two parts. In the first part, we build one instance with one ciphertext. That is, we extract features of one instance from only one ciphertext sequence. To eliminate the effect of class imbalance, we only use 1000 ciphertexts for each class. The rule is very simple: for Reuters21578 and MajorMiner, we choose the largest 1000 documents. For Caltech 101 dataset, we choose the largest 1000 images in "motorcycle" and "airplane" categories. In the second part, each instance is built to contain multiple ciphertexts, as we want to see if machine-learning algorithms can perform better by using more types of information, e.g., positions in ciphertext sequences. Each ciphertext is generated by randomly picking a plaintext from dataset (with replacement); a random IV also needs to be picked if CBC mode is used. The block ciphers used below are Data Encryption Standard (DES) and Advanced Encryption Standard (AES), where 128-bit version of AES is used. Besides, the result generated from the stream cipher RC4 is also included. In all experiments, a fixed random key is used for each cipher. In each experiment, the datasets are divided into 5 parts, and we repeatedly use four of them as the training data while the remaining one as the testing data. We use cross-validation to find the best linear solver and parameters for each part, and the final results are the average of the 5 testing data parts.

The main classifiers used are linear solvers in LIBLINEAR [18], including L2-regularized L2-loss support vector classification (dual), L2-regularized L1-loss support vector classification (dual), L1-regularized L2-loss support vector classification, and L2-regularized logistic regression (dual). The linear classifiers are very fast and suitable for bag-of-words model. For some experiments, SVM with Gaussian kernel is also used to deal with small number of features via LIBSVM [19].

We use OpenSSL[1] as our encryption tool, which is open-source and designed originally for the SSL/TLS protocol implementation. The random IVs are generated by Mersenne twister, a sophisticated pseudo random number generator [20].

3.2 Features

We list the features we use in Table 1. The first two features are related to entropy, which are calculated on a per 16- and 12-bit symbol basis, respectively. A simple scaling has been done on the entropy features via divided by the maximum entropy. The third and fourth features are the number of symbols appearing in the ciphertext. Here 2 features are extracted, and the numbers are scaled via divided by the maximum possible number of symbols as well. The fifth, sixth, seventh, and eighth features are 16-bit histograms with 65536 dimensions. The difference is that a different number of bits in the preceding ciphertext segment are XORed with the current segment to reflect the block lengths of DES and AES. Furthermore, although 8-bit histogram is used in Sharif's work, we found that it contains no useful information, as each bin has almost the same probability in our datasets. The ninth feature is the varying length words proposed by Dileep. By choosing the four most frequently appearing 4-bit delimiters, we can derive a varying length word representation. However, we note that the fixed length word representation proposed by Dileep

[1] http://www.openssl.org

is not useful in our datasets because each word appears at nearly the same frequency.

Table 1. The list of features used in experiments

Feature	Dimension	Notation
Entropy (1 symbol = 16 bits)	1	ENT1
Entropy (1 symbol = 12 bits)	1	ENT2
Number of 16-bit symbols	1	NSYM1
Number of 12-bit symbols	1	NSYM2
16-bit histogram	65536	HIST
XORed with previous 16 bits and build 16-bit histogram	65536	XOR1
XORed with previous 64 bits and build 16-bit histogram	65536	XOR2
XORed with previous 128 bits and build 16-bit histogram	65536	XOR3
Varying length words	Varies with data	VLW
Distribution of intervals between 0x00	Varies with data	INT
Ratio of zero in i-th byte, i=1…128	128	ZRO_RATIO
Entropy of the i-th byte, i=1…128	128	ENT_BYTE

The tenth feature is inspired by the varying length words representation. The idea is to use only one delimiter, so we can record the length of interval between two delimiters.

3.3 Experiment Result

Table 2 shows the results of entropy-related features proposed by Manjula, in which results labeled with RBF are obtained using SVM with Gaussian kernel. Only Reuter21578 datasets can be partially classified with just 4 features in ECB mode. We believe the main reason is that the block sizes of AES and DES are not equal, and naturally the ciphertexts produced by AES tend to have higher entropy because it uses larger blocks.

Besides, the content or size of plaintexts may implicitly affect the entropy. For example, some of documents in Reuters21578 have similar titles (No. 15871 and No. 15875), and some of the images in Caltech101 also have the same headers because their resolution is the same. For WAVE files, the results are not as strong. Our reasoning goes as follows. Assume two plaintext messages have one same block in the beginning, but other bits are totally different and random. Then the entropy should increase and approach maximum as the message size increases, resulting in poorer performance in classifying larger WAVE files.

Table 3 shows the results of histogram-related features. The cipher used can be identified in all 3 datasets in ECB mode. It is consistent with the results obtained in Dileep's and Sharif's works. However, if CBC mode is used, and if different IVs are used to produce ciphertexts, then the resulting accuracy becomes close to 50%, i.e., no better than coin flipping. This is because CBC mode can eliminate repeated patterns in ciphertexts. Besides, in the three bottommost rows, we try all 3 datasets with the same cipher but different modes of operation as labeled. Two of them can be classified with 100% accuracy, while image data has only 67.05% accuracy. There are two possible reasons. (1) A JPEG image consists of multiple segments, each of which begins with a marker[2]. Hence, the positions of one marker may vary in different files. (2) JPEG is a compressed format, which has higher entropy than uncompressed formats like text files. Nevertheless, the overall results of classification based on modes of operation are still quite acceptable.

We also try the varying length words feature (in Table 4), originally introduced by Dileep. The dictionary is directly built from the instances we used. In summary, 949540 words are found from Reuters21578, while 3449174 words are found from Caltech101, but this feature still does not help anymore in CBC mode. As AES has passed some standard NIST randomness tests [21], we further propose several randomness-related features not included in the NIST tests. The classification results are in Table 5, which shows that the accuracy is still around 50%. Therefore, the existing features do not seem to be effective in this scenario. The results of the case that an instance contains multiple ciphertexts are listed in Table 6. The term "bagsize" refers to the number of ciphertexts included in one instance. From the table, we found the accuracy tends to be around 50% as the bag size increases.

Table 2. Classification results of entropy-related features

Datasets	Ciphers	Features	Modes of operation	Accuracy
Reuters21578	AES vs. DES	ENT1+ ENT2+ NSYM1+ NSYM2	ECB	74.10%
				80.20% (RBF)
Reuters21578	AES vs. DES	ENT1+ ENT2+ NSYM1+ NSYM2	CBC	49.3%
				48.00% (RBF)
Caltech101	AES vs. DES	ENT1+ ENT2+ NSYM1+ NSYM2	ECB	51.45%
				53.94% (RBF)
Caltech101	AES vs. DES	ENT1+ ENT2+ NSYM1+ NSYM2	CBC	50.05%
				48.49% (RBF)
MajorMiner	AES vs. DES	ENT1+ ENT2+ NSYM1+ NSYM2	ECB	50%
				49.80% (RBF)
MajorMiner	AES vs. DES	ENT1+ ENT2+ NSYM1+ NSYM2	CBC	50%
				49.65% (RBF)

[2] http://class.ee.iastate.edu/ee528/Reading%20material/JPEG_File_Format.pdf

Table 3. Classification results of histogram-related features

Datasets	Ciphers	Modes of operation	Accuracy
Reuters21578	AES vs. DES	ECB	100%
Reuters21578	AES vs. DES	CBC	51.05%
Caltech101	AES vs. DES	ECB	100%
Caltech101	AES vs. DES	CBC	49.95%
MajorMiner	AES vs. DES	ECB	100%
MajorMiner	AES vs. DES	CBC	50%
Reuters21578	AES	CBC vs. ECB	100%
Caltech101	AES	CBC vs. ECB	67.05%
MajorMiner	AES	CBC vs. ECB	100%

Table 4. Classification results of varying length words features.

Datasets	Ciphers	Features	Modes of operation	Accuracy
Reuters 21578	AES vs. DES	VLW	CBC	49.05%
Caltech 101	AES vs. DES	VLW	CBC	49.55%

Even for RC4, which has been shown to have biased outputs in the second byte 錯誤！找不到參照來源。, we still cannot distinguish it from AES, as is evident from the fact that accuracy is still around 50%. It shows that more training data or a larger bag size might be required.

4. DISCUSSION AND CONCLUSION

Our experiments show that the difficulty of this task may varies with type of plaintexts, size of documents, and the modes of operation used to encrypt. Several existing features are used to predict ciphers when different modes of operation, ciphers, or types of plaintexts are given. We found that the existing features are still not capable of distinguishing encryption algorithms in the scenario in which CBC mode is used with different IVs assigned to each ciphertext. In fact, random IV is also an important factor in this problem. For example, if only one fixed IV is assigned for every ciphertext produced by a fixed secret key, then those plaintexts with the same header must be encrypted in the same manner, and the contents of first block will be the same as well. Therefore, the classification task would be a little bit easier. Since the IVs are seldom the same in real world applications, this task is still very hard and challenging today.

Overall, we find that state-of-the-art machine learning techniques are not yet effective for identification of encryption algorithm used given only a reasonably large number of sample ciphertexts. Despite that there have been successful reports in the literature, our experiments show that these works are flawed in the sense that they didn't consider CBC mode of operation with random IV, which is the recommended configuration capable of providing the basic level of security. Perhaps more advanced machine learning techniques could be applied in this problem, but we suggest that researchers must use ciphers in CBC or similar mode with a random IV in the future.

Table 5. Classification results of histogram-based features constructed from XORed segments and intervals between the delimiter '0x00'

Datasets	Ciphers	Features	Modes of operation	Accuracy
Reuters21578	AES vs. DES	XOR1	CBC	49.10%
Caltech101	AES vs. DES	XOR1	CBC	49.15%
MajorMiner	AES vs. DES	XOR1	CBC	50%
Reuters21578	AES vs. DES	XOR2+ XOR3	CBC	51.05%
Caltech101	AES vs. DES	XOR2+ XOR3	CBC	49.45%
MajorMiner	AES vs. DES	XOR2+ XOR3	CBC	50%
Reuters21578	AES vs. DES	INT+XOR1	CBC	52.55%
Caltech101	AES vs. DES	INT+XOR1	CBC	48.90%

Table 6. Classification results using multiple ciphertexts encrypted in CBC mode

Datasets	Ciphers	Features	Bagsize	Accuracy
Reuters 21578	AES vs. DES	ZRO_RATIO + ENT_BYTE	100	48.10%
Reuters 21578	AES vs. DES	ZRO_RATIO + ENT_BYTE	200	50%
Caltech 101	AES vs. DES	ZRO_RATIO + ENT_BYTE	100	49.35%
Caltech 101	AES vs. DES	ZRO_RATIO + ENT_BYTE	200	50.25%
MajorMiner	AES vs. DES	ZRO_RATIO + ENT_BYTE	100	49.55%
MajorMiner	AES vs. DES	ZRO_RATIO + ENT_BYTE	200	50%
Reuters 21578	AES vs. RC4	ZRO_RATIO + ENT_BYTE	100	49.90%
Reuters 21578	AES vs. RC4	ZRO_RATIO + ENT_BYTE	200	50%
Caltech 101	AES vs. RC4	ZRO_RATIO + ENT_BYTE	100	49.30%
Caltech 101	AES vs. RC4	ZRO_RATIO + ENT_BYTE	200	50.05%
MajorMiner	AES vs. RC4	ZRO_RATIO + ENT_BYTE	100	50.40%
MajorMiner	AES vs. RC4	ZRO_RATIO + ENT_BYTE	200	50.10%

5. ACKNOWLEDGMENTS

This work was supported in part by National Science Council, National Taiwan University and Intel Corporation under Grants NSC 100-2911-I-002-001, and 101R7501

6. REFERENCES

[1] Itsik Mantin and Adi Shamir (2001). A Practical Attack on Broadcast RC4. *FSE*, pp152 – 164.

[2] Martin R. Albrecht, Kenneth G. Paterson, and Gaven J. Watson (2009). Plaintext Recovery Attacks against SSH. *IEEE Symposium on Security and Privacy*, pp. 16–26.

[3] R. Spillman, M. Janssen, B. Nelson, and M. Kepner (1993). Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers. *Cryptologia*, vol. 17, no. 1, pp. 31–44.

[4] R. A. J. Matthews (1993). The use of genetic algorithms in the cryptanalysis. *Cryptologia*, vol. 17, no. 4, pp. 187–201.

[5] R. Spillman (1993). Cryptanalysis of knapsack ciphers using genetic algorithms. *Cryptologia*, vol. 17, no. 4, pp. 367–377.

[6] A. M. B. Albassal and A-M. A. Wahdan (2004). Genetic algorithm cryptanalysis of a Feistel type block cipher. In *proceedings of IEEE International Conference on Electrical, Electronic and Computer Engineering (ICEEC'04)*, pp. 217–221.

[7] Z. Ramzan (1998). On using neural networks to break cryptosystems. Technical report, Laboratory of Computer Science, Massachusetts Institute of Technology. Cambridge, MA 02139.

[8] A. M. B. Albassal and A-M. A. Wahdan (2004). Neural network based cryptanalysis of a Feistel type block cipher. In *proceedings of IEEE International Conference on Electrical, Electronic and Computer Engineering*, ICEEC'04, pp. 231–237.

[9] Pooja Maheswari (2001). Classification of ciphers. M. Tech Thesis. Department of Computer Science and Engineering, Indian Institute of Technology. Kanpur.

[10] Girish Chandra. The classification of modern ciphers (2001). M. Tech Thesis. Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur.

[11] A. Dileep and C. Chandra Sekhar (2006). Identification of Block Ciphers using Support Vector Machines. *International Joint Conference on Neural Networks* Vancouver, Canada, pp. 2696-2701.

[12] G. Saxena (2008). Classification of Ciphers using Machine Learning. Master's thesis, Department of Computer Science and Engineering, Indian Institute of Technology. Kanpur.

[13] Suhaila O. Sharif, L.I. Kuncheva, S.P. Mansoor (2010). Classifying encryption algorithms using pattern recognition techniques. *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on* , pp.1168-1172.
doi: 10.1109/ICITIS.2010.5689769

[14] R. Manjula and R. Anitha (2011). Identification of Encryption Algorithm Using Decision Tree. *Advanced Computing Communications in Computer and Information Science 2011* Volume 133, Part 3, 237-246.

[15] Lewis, D. D (1996). Reuters-21578 Text Categorization Test Collection Distribution. In AT&T Labs – Research.

[16] Li Fei-Fei, Rob Fergus, and Pietro Perona (2007). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*. 106, 1 (April 2007), 59-70. DOI=10.1016/j.cviu.2005.09.012 http://dx.doi.org/10.1016/j.cviu.2005.09.012

[17] M. I. Mandel and D. P. W. Ellis (2008). A web-based game for collecting music metadata. *Journal of New Music Research*, vol. 37, no. 2, pp. 151–165.

[18] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research 9(2008)*, 1871-1874.

[19] C.-C. Chang and C.-J. Lin (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2, 3, Article 27 (May 2011), 27 pages. DOI=10.1145/1961189.1961199 http://doi.acm.org/10.1145/1961189.1961199

[20] M. Matsumoto, T. Nishimura (1998). Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation 8 (1): 3–30.

[21] Soto J. (1999). Randomness testing of the AES candidate algorithms, NIST IR 6390.