Block Ciphers Classification Based on Random Forest

Xinyi Hu\textsuperscript{1,}\textsuperscript{a}, and Yaqun Zhao\textsuperscript{1,2}

\textsuperscript{1}State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi, China
\textsuperscript{2}State Key Laboratory of Cryptography and Science, China
\textsuperscript{a}Corresponding author: huxinyi.1994@foxmail.com

Abstract. Discriminant analysis is an important analytical method in cryptanalysis. When only known ciphertext, the encryption algorithm for identifying and classifying is an important part of distinguishing analysis. In this paper, a random forest classifier is used to classify eight block ciphers of ECB mode and eight block ciphers of CBC mode. By designing a feature based on ciphertext recombination and location-specific, the results show that the designed features can effectively extract the information of ciphertext data. In ECB mode, 8 algorithms can be successfully classified with an accuracy of more than 87%. In CBC mode, it can also be classified with higher accuracy than random.

1. Introduction

Many existing cryptanalysis techniques are designed for certain encryption algorithms. Therefore, in the face of big data, how to classify encryption algorithms is an important task for cryptanalysis. At the same time, the ability to resist the encryption algorithm can also be used as a measure of the security of algorithms, providing a positive reference for the design of algorithms. Most of the existing methods of classification are based on statistics and machine learning. The methods based on statistics mainly classify and identify related statistical indicators such as the frequency of occurrence of letters. The machine learning method equates the task of classifying algorithms with other common classification tasks. Most of the existing researches are conducted using machine learning methods.

In 2006, Dileep et al. \cite{1} proposed a classification scheme based on Support Vector Machine (SVM) for five block ciphers (AES, DES, 3DES, Blowfish and RC5) in ECB mode. In 2011, Manjula et al. \cite{2} proposed a classification scheme based on Decision Tree for 11 encryption algorithms including classical ciphers, stream ciphers, block ciphers, and public key ciphers. In 2013, Mishra et al. \cite{3} proposed a classification scheme based on C4.5 decision tree for AES, DES and Blowfish algorithms, using block length and entropy as features. In 2016, Tan et al. \cite{4} proposed an SVM-based classification scheme for 64-bit Blowfish, 3DES, RC5, DES and 128-bit AES algorithms. In 2017, Ray et al. \cite{5} used Fisher’s discriminant analysis for ciphertext classification for the first time. The ciphertexts of five simplified sequence ciphers and five simplified block ciphers were classified by three extracted features. In 2018, Huang et al. \cite{6} used Random Forest to conduct a two-stage identification scheme of 42 algorithms including classical ciphers, stream ciphers, block ciphers, and public key ciphers. In 2018, Tan et al. \cite{7} used SVM to classify five kinds of block ciphers in CBC mode, which are AES, DES, 3DES, RC5 and Blowfish.

In this paper, Random Forest was used to classify 16 kinds of block ciphers by designing a new feature based on bytes and frequencies of recombined ciphertext, which are eight block ciphers AES-128, AES-256, Blowfish-64, Camelia-128, DES-56, 3DES-56, IDEA-64, and SMS4-128 in ECB.
mode and these eight in CBC mode. The results show that in ECB mode, the system can classify the ciphertext data encrypted by the eight algorithms with an average accuracy of close to 88%. And in CBC mode, different algorithms can be classified by higher than random results.

Section 2 introduces the working mode of the block ciphers and the principle of the classifier. Section 3 proposes a new feature, Section 4 establishes the classification system, Section 5 shows experiments and results, and Section 6 gives the conclusion.

2. Methodology

2.1 Working Mode of Block Cipher

Block cipher is one of the important encryption systems in modern cryptography. Its main task is to provide data confidentiality. The encryption method of block cipher is that plaintext blocks are encrypted as a whole and usually ciphertext blocks of the same length as the plaintext block would be obtained.

The input of the block cipher are plaintext blocks having a fixed length of b-bit and a key, and output the ciphertexts of b-bit. If the length of the plaintexts is greater than b, it can be simply divided into blocks of b-bit.

Encrypting multiple blocks with the same key each time would lead to many security issues. In order to apply block ciphers to a wide variety of practical applications, NIST defines five working modes. Essentially, the working mode is a technique that enhances the algorithms or adapts the algorithms to a specific application, such as applying a block cipher to a sequence or stream of data blocks. And these five modes actually also cover a large number of applications of block ciphers. The five working modes are electronic-codebook (ECB) mode, cipher-block chaining (CBC) mode, Cipher Feedback (CFB) mode, output feedback (OFB) mode and counter (CTR) mode. The two most commonly used modes are ECB mode and CBC mode.

The ECB mode is a simple working mode of block cipher. The ciphertext is divided into several blocks according to the block length before encryption, and then each block is separately encrypted using the same key, and the decryption is the same. Figure 1 is the encryption process of the block cipher in ECB mode.

![Figure 1. Flow chart of block cipher in ECB mode.](image)

This encryption method is simple and easy to operate, and can be parallelized to shorten the encryption time. However, since the same ciphertext blocks are encrypted by the same plaintext blocks, it cannot hide the data pattern very well. It is vulnerable to replay attacks in some cases and does not provide strict data confidentiality.

The CBC mode is the most commonly used mode. Each plaintext block would XOR with the previous ciphertext block and then encrypted. The first block needs to use the initialization vector IV. Each ciphertext block depends on all plaintext blocks in front of it.

![Figure 2. Flow chart of block cipher in CBC mode.](image)
The CBC mode can guarantee the uniqueness of each piece of information. When encrypting, a small change in the plaintext will cause all subsequent ciphertext blocks to change. However, its main disadvantage is that the encryption process is serial and cannot be parallelized, and the message must be padded to an integer multiple of the block length.

2.2 Random Forest
Random forest is an algorithm that integrates multiple trees through the idea of integrated learning. Its basic unit is the decision tree, and its essence belongs to the ensemble learning method, a branch of machine learning.

Random forest is a very flexible and practical method, and it has twelve characteristics [10][11], five main characteristics are as follow:

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.

The principle of random forest is based on bagging strategy. And the bagging strategy is derived from bootstrap aggregation: Suppose the sample set has \( N \) data points, \( n \) samples are resampled from the sample set, and there are returned samples. The number of sample data points remains unchanged to \( N \). On all samples, a classifier is built for the \( n \) samples. Repeat the above two steps \( m \) times, obtain \( m \) classifiers, and finally decide which class the data belongs to based on the voting results of the \( m \) classifiers.

The process of constructing a random forest is as follows:
1. Randomization of samples: Randomly select \( n \) samples from Bootstrap from the sample set.
2. Randomization of features: Randomly select \( k \) attributes from all attributes, select the best segmentation attribute as a node to establish a CART decision tree (other types of classifiers are also available, such as SVM, Logistics).
3. Repeat the above two steps \( m \) times, that is, establish \( m \) CART decision trees.
4. These \( m \) CARTs form a random forest. And by polling the results, it decides which class the data belongs to (the voting mechanism has one vote veto system, minority subject to majority, weighted majority).

3. Feature Selection and Extraction
Since the block ciphers used in this paper are used in practical applications after a large number of security tests and randomness tests. The ciphertexts obtained by these algorithms are very random, and it is difficult to classified by human beings, and is difficult to select and extract features.

Considering the internal structure of the block ciphers, since the block cipher often has different block lengths, and in ECB mode, each block is encrypted according to the same encryption method. Thus after reorganizing ciphertext, it may be possible to classify different algorithms based on ciphertext at a specific location.

The ciphertexts were separated by the length of 8 bits (1 byte), extract the frequency of the occurrence of fixed bits of each byte, and obtain the characteristic features \((f_1, f_2, \ldots, f_d)\), where \( d \) is the feature dimension.

1. The \( n \)-bit long ciphertexts are separated by bytes to obtain \( n/8 \) bytes, and the first bit of each byte is extracted to obtain ciphertext \( c^1, c^2 \ldots, c_{n/64}^1 \). The length of \( c^1 \) is \( n/64 \) bytes, that is \( n/8 \) bits.
2. Record the occurrence frequencies of the \( n/64 \) bytes \( c_1^1, c_2^1, \ldots, c_{n/64}^1 \) in \( c^1 \) as the first extracted feature.
3. The 2nd bit, the 3rd bit, the 8th bit of each byte of the ciphertexts are sequentially extracted, and the corresponding ciphertexts are obtained, \( c^2 = \{c_1^2, c_2^2, \ldots, c_{n/64}^2\}, c^3 = \{c_1^3, c_2^3, \ldots, c_{n/64}^3\}, \ldots, c^8 = \ldots, c^d = \ldots \)
\{c_1^8, c_2^8, ..., c_{n/64}^8\}. Record the occurrence frequencies of \(c_i^j\) \((i = 1, 2, ..., n/64, j = 1, 2, ..., 8)\) in each byte.

There are \(2^8\) possible cases for one byte, and the number of occurrences of each case is recorded. The feature of \(2^8=256\) dimensions can be obtained by one extraction, and 8 times are extracted, so the final dimension is \(256\times 8=2048\).

4. Classification Scheme Based on Random Forest

According to the classifier in Section 2 and the features in Section 3, a cryptanalysis classification system based on random forest was established. The flow chart is shown in Figure 3.

![Flow chart of classification scheme based on random forest.](image)

**Figure 3.** Flow chart of classification scheme based on random forest.

**Training phase:**
1. The ciphertext files \(F_1, F_2, ..., F_m\) have been known the classes, where \(m\) is the number of files.
2. Feature extraction is performed on the ciphertext files to obtain the feature \(\text{Fea} = \{\text{fea}_1, \text{fea}_2, ..., \text{fea}_m\}\), where \(\text{fea}_i\) \((i = 1, 2, ..., m)\) are vectors with a dimension of 2048.
3. Tag labels of \(m\)-dimensional vectors \(\text{Lab} = (l_1, l_2, ..., l_m)\) for \(m\) ciphertext files of known classes, \(l_i\) \((i = 1, 2, ..., m)\) represent \(k\) different algorithms, respectively. Then get a set of tagged data \((\text{Fea}, \text{Lab})\).
4. Put \((\text{Fea}, \text{Lab})\) into the random forest classifier for training.

**Test phase:**
1. Extract the feature \(\text{fea}^*\) of the file \(F\) which is to be classified and unknown the class.
2. Input \(\text{fea}^*\) into trained classifier, the classification result can be obtained. That is, the ciphertext file \(F\) is encrypted by the algorithm tagged \(l^*\).

5. Experiments and Results

In this section, based on the ciphertext classification system established above, the experiments are tested as follows. The experimental environment is shown in Table 1.

**Table 1.** Experiment environment.

|              |                |
|--------------|----------------|
| CPU          | 28 GB          |
| Processor    | Intel Core i7-4970 |
| Operating System | Ubuntu 18.04   |

The experiment specifically examined 16 encryption algorithms: eight block ciphers in ECB mode, AES-128, AES-256, Blowfish-64, Camellia-128, DES-56, 3DES-56, IDEA-64, SMS4-128, and the same 8 block ciphers in CBC mode.

In the data collection phase, data from the California Institute of Technology Caltech-256 dataset [8] was selected, and the data into 1001 files was stitched as plaintext, and the size of files are 512KB. Then encrypted them with the above 16 algorithms to obtain 1001 ciphertext files respectively. Cut off
these ciphertext files, the file length is still controlled to 512KB. Finally, there is a total of 16016 ciphertext files of 512KB in size, including 8008 ciphertext files of 8 algorithms in ECB mode and 8008 ciphertext files in CBC mode. The block ciphers are implemented by the open source tool OpenSSL, and the ciphertexts of each algorithm are encrypted with the same key in training phase and test phase.

The experiment examines the accuracy of block ciphers classification under the two modes based on the system in Section 4. In the experiment, for the two modes, 7200 files in 8008 ciphertext files are randomly selected as the training set, and the remaining 808 files are used as the test set. And the training set and the test set are both included in the ciphertext of 8 algorithms. Feature extraction and ciphertext classification are implemented by Python 3.6.5.

Each time the classification is repeated 10 times, the overall accuracy is equal to 10 mean values. The human interference is eliminated to the greatest extent, and the robustness of the model is confirmed. In ECB mode, the classification accuracies of the eight algorithms are shown in Figure 4 and Table 2. In CBC mode, the classification accuracies of the eight algorithms are shown in Figure 5.

![Figure 4. Classification accuracy of eight block ciphers in ECB mode.](image)

It can be seen from Figure 4 that in 10 experiments, the classification accuracies in ECB mode are stable above 86%, and the average accuracy is also close to 88%. Since the ECB mode is separately encrypted for each block using the same key, the security is not as high as other modes of operation, and it is easier to classify than others, so the accuracy is also high.

![Figure 5. Classification accuracy of eight block ciphers in CBC mode.](image)

It can be seen from Table 2 that the classification model proposed in this paper can successfully classify the eight algorithms in ECB mode with an average accuracy of 87.9%, where the percentage in parentheses indicates the percentage of random classification. Moreover, compared with the existing researches, the accuracy rate is the highest in this paper, and the results of the other existing researches are less than 50%.

| Source       | this paper | [6] | [5] | [1] | [9] |
|--------------|------------|-----|-----|-----|-----|
| Accuracy     | 87.9%      | 21.5% | 36.65% | 41% | 30.84% |
| (12.5%)      | (20%)     | (20%) | (20%) | (20%) | (12.5%) |

It can be seen from Table 2 that the classification model proposed in this paper can successfully classify the eight algorithms in ECB mode with an average accuracy of 87.9%, where the percentage in parentheses indicates the percentage of random classification. Moreover, compared with the existing researches, the accuracy rate is the highest in this paper, and the results of the other existing researches are less than 50%.
It can be seen from Figure 5 that the classification accuracies in CBC mode are much lower than those in ECB mode, the highest is no more than 14%, and the average accuracy is only 12.64%. This is because the CBC mode is more secure and less vulnerable to attack, and the ciphertext is also very random and difficult to classify. In the existing researches, there are few classification studies on the CBC model so there is no comparison with the existing researches.

6. Conclusion
This paper discusses the classification of ciphertexts generated by a total of 16 different algorithms including eight block ciphers in ECB mode and eight in CBC mode. Aiming at the research goal, the feature of frequency of specific location based on recombination ciphertext is designed. The ciphertext feature of 16 algorithms is classified into two modes by random forest classifier.

For the classification under ECB mode, the average accuracy rate can reach 87.9%, which is much higher than the random 12.5%, and much higher than the results of the existing researches. For the classification in CBC mode, because its security is higher than the ECB mode, the average accuracy is slightly higher than random, only 12.64%.

In the future research, we would design effective classification features based on the security mode such as CBC mode based on the characteristics of encryption methods. At the same time, ciphertext classification research can also carried out for stream ciphers and public key ciphers.

Acknowledgement
This work is supported by the Open project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing; National Key Research and Development Project 2016-2018 (2016YFE0100600); State Key Laboratory of Information Assurance Technology Open Fund Project (KJ-15-008); State Key Laboratory of Cryptography and Science.

References
[1] A. D. Dileep, C. C. Sekhar, International Joint Conference on Neural Networks, 2696-2701 (2006)
[2] R. Manjula, R. Anitha, International Conference on Computer Science and Information Technology, 237-246 (2011)
[3] S. Mishra, A. Bhattacharjya, IEEE International Conference on Recent Trends in Information Technology, 393-398 (2013)
[4] C. Tan, Q. Ji. IEEE International Conference on Communication Software and Networks, 19-23 (2016)
[5] P. K. Ray, S. Kant, B. K. Roy, A. Basu, Defence Science Journal 67, 59-65 (2017)
[6] L. Huang, Z. Zhao, Y. Zhao, Chinese Journal of Computers 41, 382-399 (2018)
[7] C. Tan, X. Deng, L. Zhang, Procedia Computer Science 131, 65-71 (2018)
[8] G. Griffin, A. Holub, P. Perona, California Institute of Technology, (2007)
[9] S. O. Sharif, L. I. Kuncheva, S. P. Mansoor, IEEE International Conference on Information Theory and Information Security, 1168-1172 (2010)
[10] L. Breiman, A. Cutler, https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#inter
[11] L. Breiman, Machine Learning 45, 5-32 (2001)