Artificial Intelligence Applied to the Identification of Block Ciphers under CBC Mode

Bruno dos S. Rocha
Military Institute of EngineeringRio de Janeiro, RJ, Brazil

José A. M. Xexéo
Military Institute of EngineeringRio de Janeiro, RJ, Brazil

Renato H. Torres
Federal University of Para´Belém, PA, Brazil

ABSTRACT

This research introduces a novel methodology for identifying symmetric cryptosystems operating in Cipher Block Chaining (CBC) mode based solely on encrypted texts. The approach combines statistical tests from NIST STS with machine learning algorithms, analyzing DES, 3DES, Blowfish, Camellia, and AES. The experimental results demonstrate an 84% identification rate for multiclass identification using random keys and initialization vectors. These findings are valuable in the field of information security and aid in minimizing cryptanalytic efforts.

Keywords

Identification, Block cipher, CBC mode, NIST STS, Machine Learning

1. INTRODUCTION

The precise identification of cryptographic algorithms from their ciphertexts is crucial because, as stated by Cheng Tan et al. [1], a comprehensive understanding of the algorithm employed is essential in all cryptanalytic scenarios.

The literature presents research dedicated to the identification of cryptographic algorithms through patterns present in encrypted texts, using machine learning algorithms. However, the identification becomes more challenging when the Cipher Block Chaining (CBC) mode is employed, due to the high randomness resulting from the chaining of encrypted blocks. This complexity and randomness make the process of analysis and identification more intricate and challenging.

In the context of an ”Only Ciphertext” attack, this research aims to propose a methodology for identifying symmetric cryptosystems operating in CBC mode from encrypted texts with improved accuracy compared to previous studies. The analysis focuses on well- known block ciphers, including DES, 3DES, Blowfish, Camellia, and AES.

This article follows this structure: Section 1 presents the work’s context and main contributions. Section 2 reviews relevant literature. Section 3 discusses theoretical foundations, including CBC mode, symmetric cryptosystems, and machine learning algorithms. Section 4 introduces the identification method. Section 5 presents experimental results and performance analysis. Finally, Section 6 concludes the work, highlighting key findings and suggesting future research directions.

2. RELATED WORK

Data mining is essential for extracting patterns from large volumes of information [2], using algorithms to identify connections and analyze future trends, enabling class prediction [3]. In the context of cryptography, data mining can be employed to identify the cryptographic algorithm responsible for generating a ciphertext, based onthe recognition of patterns in encrypted texts. Therefore, machine learning algorithms are suitable tools for data mining in cryptography, making significant contributions to the identification of cryptographic algorithms.

Cheng Tan et al. [4] proposed a method based on Support Vector Machines (SVM) to identify AES, DES, 3DES, RC5, and Blowfishblock ciphers operating in CBC mode, using 200 encrypted text files for each cipher. Out of these 200 files, 20 were used for classifier training, and the remaining for testing. The authors explored different configurations of keys and initialization vectors (IVs), andwhen utilizing different keys and IVs in 500 KB files, the identification rate reached 38.67%. Despite being an innovative methodology, the lack of information regarding the criteria for extracting features and obtaining representative vectors from the analyzed ciphertexts hinders a comprehensive understanding of the methodology and the replication of the experiments.

In the study conducted by Fan and Zhao [5], they employed the Euclidean distance between encrypted texts generated by cryptographic algorithms such as DES, 3DES, AES-128, AES-256, IDEA, SMS, Blowfish, and Camellia-128 in CBC mode as featuresfor the Random Forest (RF), Logistic Regression (LR), and Sup- port Vector Machine (SVM) classifiers. The researchers collected data by generating 1001 samples of 512 KB for each algorithm andmode of operation, resulting in a total of 16016 encrypted text files,all encrypted with fixed keys. The method achieved an accuracy of 13.5%, surpassing random classifications, which had an accuracy rate of 12.5%. However, the authors did not provide details about the IVs used and did not specify if they were fixed or random.

In [6], Dileep and Sekhar proposed an approach based on the bag- of-words model and machine learning to identify cryptographic algorithms. The authors generated encrypted texts from a plaintext of 4000 bits, containing 500 ASCII characters. They analyzed the algorithms DES, 3DES, Blowfish, AES, and RC5, using random keys for each algorithm. The authors conducted experiments usingK-Nearest Neighbors (KNN) classifiers with k values of 5, 15, and25, as well as Support Vector Machine (SVM) classifiers with linear, polynomial, sigmoid, and Gaussian kernels. The results of the analysis showed that the SVM classifier achieved the highest accuracy rates, particularly when the Gaussian kernel was used, reaching an accuracy of 87% in the scenario where the same key was used in both the training and testing datasets, and the DES algorithm was analyzed. However, when different keys were used in the training and testing datasets, the accuracy decreased significantly to 35%.

In [7], Mello and Xexéo utilized machine learning algorithms to identify cryptographic algorithms operating in CBC mode. The experiment involved corpora of plaintext in seven different languages (Portuguese, Spanish, English, German, Hebrew, Cyrillic, and Mandarin), seven cryptographic algorithms (DES, Blowfish, RSA, ARC4, Rijndael, Serpent, and Twofish), and six machine learning classifiers (C4.5, PART, FT, Naive Bayes (NB), Multilayer Perceptron, and WiSARD). The corpora of plaintext used consistedof 4200 samples divided into seven distinct corpora, each representing a language system. Each corpus comprised 600 samples ofdifferent texts collected from newspapers and magazines, with no repeated sentences. Each encrypted file was represented by a histogram that recorded the occurrence of contiguous bit blocks, ranging from 2 to 34 bits. The results showed successful identification of the cryptographic algorithms, with results higher than random probability (13%). The most efficient classifier was the NB with anaccuracy of approximately 50%.

In [8], Hu and Zhao used the RF algorithm to classify block ciphersAES-128, AES-256, Blowfish-64, Camellia-128, DES-56, 3DES-56, IDEA-64, and SMS4-128 operating in CBC mode. Clear texts from the Caltech256 dataset were grouped into 1001 files of 512 KB and encrypted with the eight ciphers, resulting in a total of 8008encrypted files. It is important to note that the clear texts for each algorithm were encrypted with the same key both during the training and testing phases of the machine learning model. The authors then employed a dictionary-based method with 8-bit words to rep-resent the analyzed ciphertexts. The classification model achieved an average accuracy of 12.64%, slightly higher than random chance (12.5%).

The NIST Statistical Test Suite (NIST STS), introduced in NIST SP 800-22 [9], is a collection of 15 statistical tests used to assess the randomness quality of bit sequences generated by Random Number Generators (RNGs) and Pseudorandom Number Genera- tors (PRNGs) employed in cryptographic applications. These testsare designed to detect deficiencies in random bit sequences, such as patterns, dependencies, and non-randomness. Additionally, scientists and engineers in various fields, including information security, cryptography, and signal processing, also use the NIST STS for analysis and evaluation purposes.

In the study by Yu and Shi [10], a novel approach was proposedto identify ciphertexts in CBC mode. They used the results of fivetests from the NIST STS test suite to represent the ciphertexts generated by DES, AES, 3DES, and Blowfish algorithms, all used withequal keys. The encrypted texts were obtained from 4000 files of the Caltech-256 dataset, each with a size of 256 KB. They then ap-plied the MultiLayer Perceptron (MLP) classifier using 75% of the data as the training set and the remaining as the test set. The accuracy of the MLP classifier was found to be 29.8%, surpassing the random probability of 25%.

Based on this analysis, it can be asserted that the utilization of ma- chine learning is a valuable tool for identifying cryptographic sys- tems, justifying its application in this research to identify cryptographic algorithms in CBC mode. Due to the high degree of ran- domness in this mode, the development of an effective identification method holds significant value for cryptology and research focused on data security and sensitive information.

3. BACKGROUND KNOWLEDGE

3.1 Cipher Block Chaining encryption mode

According to [9], in the Cipher Block Chaining (CBC) mode of operation, the plaintext is divided into fixed-size blocks, typically 64 or 128 bits, which are encrypted sequentially, taking into account the feedback from the previously encrypted blocks. It is common to use padding schemes to adjust the message size, allowing it to be divided into blocks of the appropriate size for encryption, ensuring that the total length is a multiple of the block size. PKCS5 [11]and PKCS7 [12] are examples of widely used padding schemes inblock encryption.

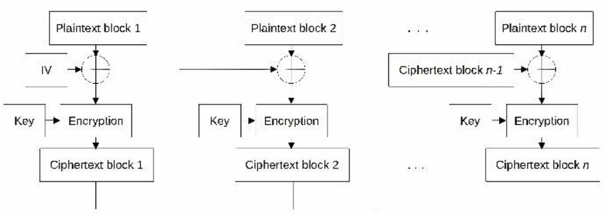

Fig. 1: The operation of CBC encryption mode

As illustrated in Figure 1, in this mode of operation, the first plain- text block is XORed (exclusive OR) with the initialization vector (IV), a sequence of random or fixed numbers with the same sizeas the plaintext block. The resulting block from the XOR operationis then encrypted to form the first ciphertext block. This cipher- text block is then combined through XOR with the second plaintext block before being encrypted, and so on for the remaining blocks. The encryption process is completed when the last ciphertext block *n* is obtained. This combination of blocks through XOR ensures that each block depends on the previous block, creating a chaining of blocks or ”cascading effect” that enhances the security of the cipher.

3.2 Block Ciphers

There is a wide variety of cryptographic algorithms, each with its own mathematical constructions and levels of security. Some of these algorithms were developed decades ago and continue to be analyzed and employed to this day.

The DES, created in the 1970s to meet the security needs of the USA, had its popularity affected due to vulnerabilities against brute-force attacks [13]. Modifications were suggested by Tanen- baum to increase its security [14], while Pfleeger and Pfleeger emphasized security derived from substitution and transposition techniques [15].

The Blowfish algorithm, proposed by Bruce Schneier in 1993 as analternative to DES, has been demonstrated to be faster and nearly as energy-efficient [16]. Moreover, it provides higher security dueto the use of larger key sizes [17].

3DES, a variant of DES, employs DES three times sequentially, using two or three different keys. This approach significantly in- creases the key space, making 3DES more resistant to brute-force attacks [18].

The Rijndael algorithm, winner of the AES competition, replaced DES and uses keys of 128, 192, and 256 bits. Currently, AES is thestandard for symmetric encryption in the United States, owing to its efficiency and high security [19], and it continues to be widely adopted.

The Camellia algorithm, developed by Mitsuru Matsui et al. [20], underwent the selection process by the Japanese government and is known for its resistance to various cryptanalytic attacks, including differential and linear cryptanalysis. Its structure based on a modified Feistel network, the use of keys of different sizes, and the non- linear transformations during the encryption process ensure data confidentiality and integrity [21].

3.3 Machine Learning classifiers

Artificial intelligence utilizes algorithms and models that enable asystem to learn from data and make predictions without the need for explicit programming. ”Classifiers” are machine learning algorithms used to classify data into classes based on features of the input data.

NB [22] is a widely used classifier for text data due to its speed and ease of implementation. However, it has significant drawbacks,such as sensitivity to imbalanced class data during model training and inadequate handling of class overlap.

RL is a widely used algorithm in the field of machine learning for multiclass classification problems [23]. Unlike Naive Bayes, whichassumes independence between features, RL considers the linear relationship between independent variables and the probability of belonging to a particular class. Its simplicity and computational efficiency make it a popular choice in many applications. Additionally, RL provides well-calibrated probabilities for classification classes, which is particularly important in classification problems.

KNN is a popular machine learning algorithm for classification [24]. It classifies a new data instance based on the classes of the nearest training instances, where ”K” represents the number of nearest neighbors considered for the classification decision. The new instance is assigned to the class most frequently occurring among the ”K” nearest neighbors. One of the main advantages of KNN is its ability to handle complex and nonlinear relationships between features and the target variable. Moreover, it does not re-quire an extensive training process as all the knowledge is stored inthe training data.

SVM is a well-known learning algorithm for its efficiency in dealing with high-dimensional and complex problems, as well as its generalization capability to new data. It seeks to find an optimal hyperplane that separates different data classes as widely as possible,maximizing the margin between classes. The data points closest tothis hyperplane, known as support vectors, are crucial for model construction and directly influence the decision boundary [25].

RF is an algorithm based on the concept of ensemble learning, which combines multiple decision trees for classification and regression [26]. Each tree is trained on a random sample of the orig-inal data and uses only a random subset of the available attributes.The combination of these trees results by majority voting creates a more robust model less susceptible to overfitting. RF performs wellwith high-dimensional data and a large number of features, without requiring complex preprocessing. Compared to a single decision tree, it is less sensitive to overfitting.

4. PROPOSED METHOD FOR IDENTIFYING BLOCK CIPHERS IN CBC MODE

The aim of this research is to identify cryptographic systems operating in CBC mode through their corresponding ciphertexts using data mining. The proposed methodology relies on the p-values re-turned by the fifteen individual tests of the NIST STS and machine learning algorithms. This methodology was applied to the widely used and analyzed algorithms DES, 3DES, Blowfish, Camellia, and AES, with the purpose of comparing the results with other related research.

For the classification of the analyzed ciphers, machine learning algorithms LR, KNN, SVM, RF, and NB were employed. This choice was based on their widespread usage in related works and their dis-tinct approaches and characteristics, which allow for a comprehen sive analysis of the ciphertexts generated by the mentioned crypto-graphic systems.

For each ciphertext, only its first block was selected. Then, multiple first blocks from different ciphertexts were concatenated to create asingle ciphertext file. From this file, features were extracted, which were expressed as a feature vector.

Following the procedures illustrated in Figures 2 and 3, a corpus of English texts from various distinct literary genres was encryptedusing the DES algorithm, 3DES, Blowfish, Camellia, and AES with random keys and IVs, resulting in multiple ciphertexts. For each generated ciphertext, only the first block produced was selected, with a size of 128 bits for AES and Camellia algorithms, and 64 bits for DES, 3DES, and Blowfish algorithms.

These first blocks, corresponding to each of the analyzed ciphers, were concatenated into a single file containing multiple concate- nated first blocks. This procedure was replicated 100 times for eachanalyzed cipher, generating 100 different files of concatenated first blocks, resulting in a total of 500 files generated by the five algorithms.

Next, the 500 files of concatenated first blocks were subjected to the 15 statistical tests of the NIST STS, and the returned p-values were used to generate representative vectors of the analyzed files, containing the features of the ciphertexts. These representative vec- tors were then analyzed by the machine learning algorithms mentioned earlier, enabling pattern recognition and identification of the different block ciphers analyzed.

This research adopted an exploratory approach, utilizing a corpus of English texts for analysis. According to Mello and Xexeo [7], different languages did not impact the data mining process nor were they relevant for classification. To eliminate possible biases, each plaintext was encrypted with a unique key to prevent cryptographickey reuse from influencing the analysis of machine learning algorithms [3].

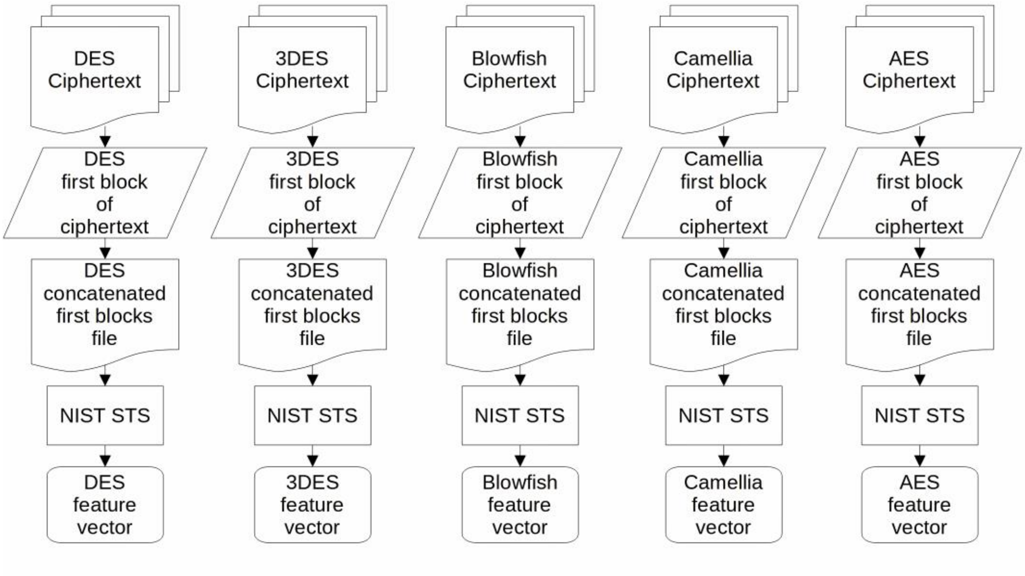
DES Ciphertext
3DES Ciphertext
Blowfish Ciphertext
Camellia Ciphertext
AES Ciphertext
DES first block of ciphertext
3DES first block of ciphertext
Blowfish first block of ciphertext
Camellia first block of ciphertext
AES first block of ciphertext
DES concatenated first blocks file
3DES concatenated first blocks file
Blowfish concatenated first blocks file
Camellia concatenated first blocks file
AES concatenated first blocks file
NIST STS
NIST STS
NIST STS
NIST STS
NIST STS
DES feature vector
3DES feature vector
Blowfish feature vector
Camellia feature vector
AES feature vector

Fig. 2: Construction of cryptosystem feature vectors

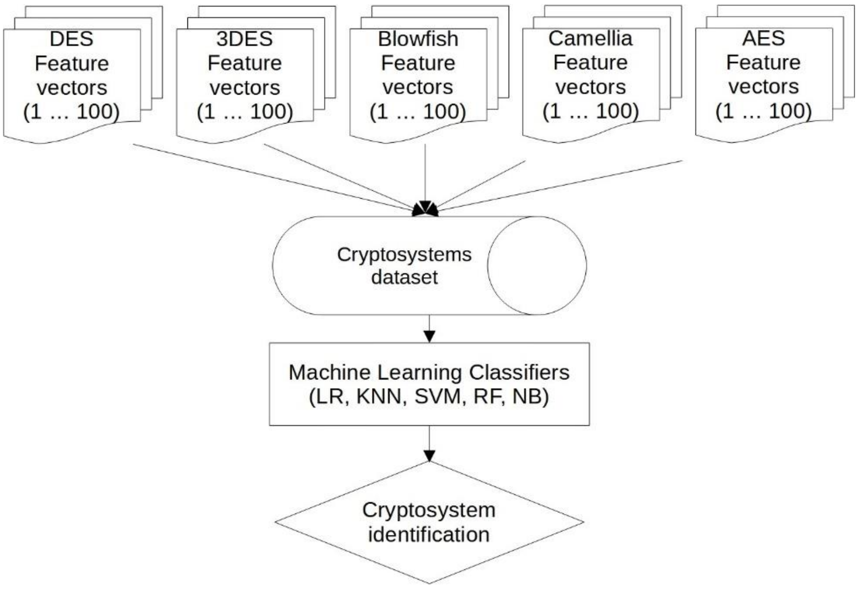
DES Feature vectors (1 … 100)
3DES Feature vectors (1 … 100)
Blowfish Feature vectors (1 … 100)
Camellia Feature vectors (1 … 100)
AES Feature vectors (1 … 100)
Cryptosystems dataset
Machine Learning Classifiers (LR, KNN, SVM, RF, NB)
Cryptosystem identification

Fig. 3: Cryptosystem identification scheme

The 500 files of concatenated first blocks were subjected to the 15statistical tests of the NIST STS, and the returned p-values were used to generate representative vectors of the analyzed files, containing the features of the ciphertexts. These representative vectors were then analyzed by the machine learning algorithms mentioned earlier, enabling pattern recognition and identification of the different block ciphers analyzed. To experiment with the proposed methodology, 100 files of concatenated first blocks were created for each cryptographic system, each with a size of 100 KB.

Afterward, the open-source tool NIST STS *sp800 22 tests-master*,previously used in the researches [27] and [28], was employed to analyze the files of concatenated first blocks. The application of NIST STS results in a set of samples where each corresponds to a vector containing the results of the 15 statistical tests of the NIST STS.

During the data preprocessing phase, the dataset was split into a training set (70% of the data) and a test set (30% of the data). Then, the trained models were tested on the test set using common evaluation metrics such as accuracy, precision, recall, and F1-score to measure the effectiveness of the machine learning algorithms.

It is important to emphasize that the same number of samples was generated for all analyzed cryptographic algorithms, ensuring im- partiality in the identification of one algorithm compared to anotherand eliminating potential biases.

5. EXPERIMENTS AND PERFORMANCE EVALUATION

The experiments were conducted on a computer with an Intel® Core™ i5-2450M CPU @ 2.50GHz × 4 processor and 6.0 GiBof RAM, running the 64-bit Ubuntu 22.04 LTS operating system. The metadata sets were processed using the Scikit-Learn machine learning platform, which provides a wide variety of machine learn-ing algorithms.

The LR classifier used the *solver* hyperparameter set to "newton-cg". The KNN was employed with the hyperparameters*n neighbors=5* and *weights='uniform'*, using the Euclidean dis- tance as the distance metric. In the case of the SVM, the hyper- parameters *C=1.0*, *kernel='rbf'*, and *gamma='scale'* were used. NB does not require specific hyperparameters for fitting, and the RF was configured with 100 estimators (*n estimators=100*) and a maximum depth of 5 in the trees (*max depth=5*).

The results of each classifier are expressed in terms of accuracy, precision, recall, and f1-score. TP (True Positive) represents the number of examples correctly classified as positive, while TN (True Negative) represents the number of examples correctly classified as negative. FP (False Positive) represents the number of examples in-correctly classified as positive, and FN (False Negative) represents the number of examples incorrectly classified as negative.

Accuracy measures the proportion of correctly classified samples out of the total dataset. Precision measures the proportion of true positives among the samples classified as positive, while recall refers to the proportion of true positives out of all positive samples. The f1-score is a metric that combines precision and recall into a single value, providing an overall measure of classifier performance. The closer the f1-score value is to 1 (or 100%), the better the classifier's performance in terms of precision and recall, indicating a good balance between these two metrics. The formulas for these metrics are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2 * (Precision * recall)}{Precision + recall}$$

The classification results can be visualized in the confusion matrices shown in Figures 4, 5, 6, 7, and 8, as well as in the graph in Figure 9, which summarizes the performance of machine learning algorithms in the classification task. In the confusion matrices, the vertical axis represents the correct classes of the samples, while the horizontal axis displays the classes predicted by the classifier. For example, in Figure 5, the KNN classifier correctly identified 12samples as Blowfish but misclassified 16 samples, with 14 being incorrectly labeled as 3DES and 4 as AES.

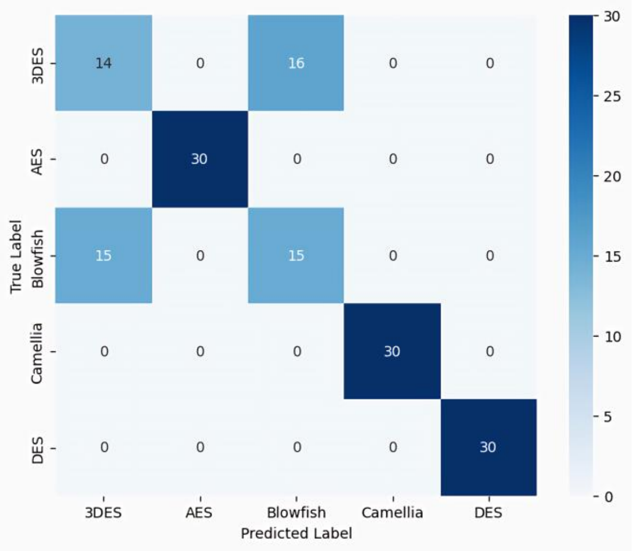

Fig. 4: LR Confusion Matrix

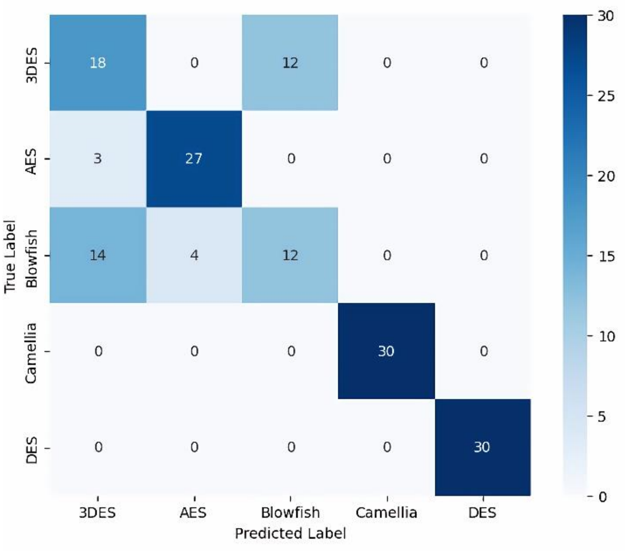

Fig. 5: KNN Confusion Matrix

The LR achieved an accuracy of 79%, indicating a robust performance. This model attained 100% precision and recall for the AESand DES algorithms. However, its performance was more moderate when classifying 3DES and Blowfish, with precision, recall, and F1 scores hovering around 50%. It is noteworthy that all samples of theCamellia algorithm were correctly classified.

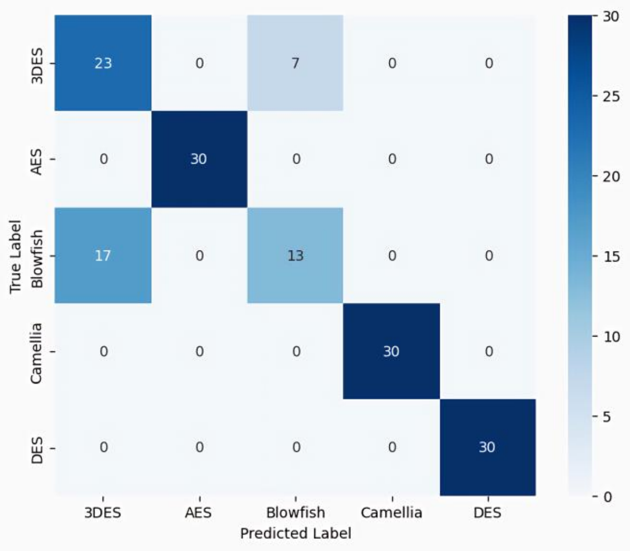

Fig. 6: NB Confusion Matrix

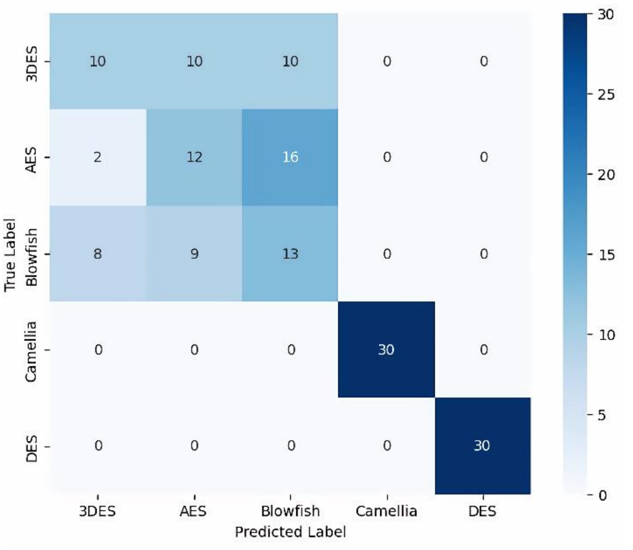

Fig. 7: SVM Confusion Matrix

The KNN model displayed an accuracy of 78%. This model outperformed logistic regression in the classification of 3DES, AES, andCamellia, with higher precision, recall, and F1 scores. However, itencountered difficulties when classifying samples of the Blowfishalgorithm, achieving scores around 50%, and performed similarly to LR in classifying the DES algorithm.

NB exhibited an accuracy of 84%. In the classification of 3DES, it achieved a precision of 57% and recall of 77%, resulting in an F1 score of 66%. Similar to the other classifiers, it showed solid performance in classifying AES and DES. However, its performance for the Blowfish algorithm was satisfactory but inferior to that observed for 3DES.

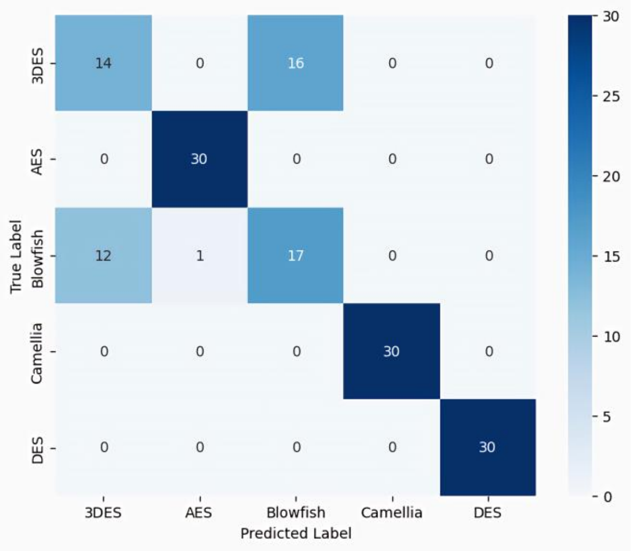

Fig. 8: RF Confusion Matrix

The SVM was the classifier with the lowest accuracy, reaching 63%. This model faced challenges when classifying samples fromthe 3DES, AES, and Blowfish algorithms, as indicated by the precision, recall, and F1 score metrics hovering around 40%. However,like the other models, SVM correctly classified all samples of the Camellia and DES algorithms.

The RF achieved an accuracy of 81%, indicating robust performance. It demonstrated good precision, recall, and F1 scores for 3DES, AES, and Blowfish, with a remarkable precision of 97% forAES. Similarly to the other classifiers, all samples of the Camelliaand DES algorithms were accurately classified.

The experiment results confirm the hypothesis that patterns exist within the ciphertexts, revealing a signature associated with the types of cryptographic algorithms. Furthermore, they reinforce the concept that the p-values reflect distinctive statistical characteristics of the cryptosystems, providing a means for characterization and differentiation. The detection of such patterns in the analyzed ciphertexts may indicate vulnerabilities susceptible to cryptanalyticattacks.

Given that five cryptographic algorithms were evaluated, each withrandom keys and initialization vectors, with no repetitions, which impart a high degree of randomness to the encrypted texts and make the analysis challenging, and considering that the probability of random guessing is 20%, the outcomes obtained with the method proposed in this study corroborate its effectiveness in identifying cryptographic algorithms operating in CBC mode. It is evident that the classifiers' accuracy significantly surpasses the random guessing rate. This consistency suggests that the proposed method is capable of handling various cryptographic algorithms while maintaining reliable and predictable performance when a large volume of encrypted text data is available to the researcher.

6. CONCLUSION

In a scenario of ciphertext-only attack, where the cryptanalyst hadlimited information, it was crucial to at least know the crypto- graphic algorithm used for encryption. Although breaking an algorithm was not a simple task, knowledge of the algorithm used could significantly reduce the effort required to obtain the originalmessage through cryptanalysis [29].

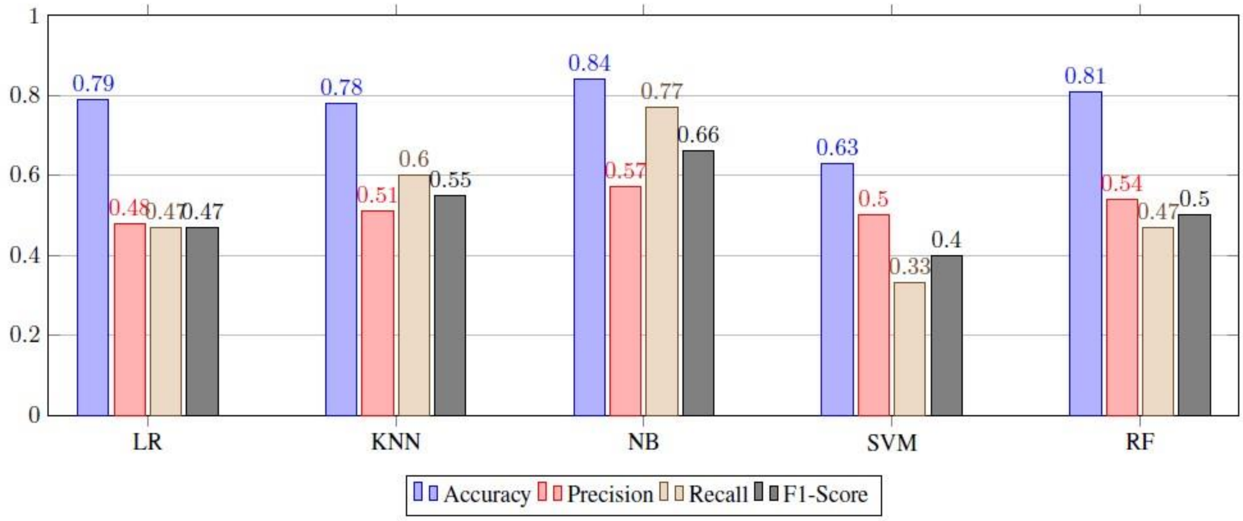

Fig. 9: Results of Experiments

In this study, the DES, 3DES, Blowfish, Camellia, and AES algorithms were examined while operating in CBC mode. The results demonstrated that, using machine learning algorithms and statistical tests, an identification rate of 84% was achieved. Patterns wereidentified in the ciphertexts generated by these systems in CBC mode, enabling attacks with ciphertext through the combination ofmachine learning algorithms and static tests from the NIST STS battery.

The methodology proposed in this research, by focusing on the initial concatenated blocks, eliminated the impact of block chaining in CBC mode. This allows researchers to identify patterns in encrypted texts and determine the algorithm responsible for generating the ciphertext when a large collection of encrypted files is available.

The robust results confirmed the effectiveness of this approach, even with random keys and initialization vectors. It outperformed related work in cipher identification, with lower computational costs compared to other mentioned techniques.

For future work, it would be interesting to explore the reduction ofthe concatenated file size of the first block to determine the limits at which cryptographic systems could be correctly classified. Additionally, including files of other types, such as images, videos, and audio, and incorporating more cryptographic algorithms could further expand the findings obtained in this study. The exploration ofalternative classifiers could also enhance the analysis and improvethe performance of this methodology in future identification tools.

7. REFERENCES

[1] Cheng Tan and Qingbing Ji. “An approach to identifying cryptographic algorithm from ciphertext”. In: *2016 8th IEEE International Conference on Communication Soft- ware and Networks (ICCSN)*. IEEE. 2016, pp. 19–23.

[2] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. “From data mining to knowledge discovery in databases”. In: *AI magazine* 17.3 (1996), pp. 37–37.

[3] Flavio Luis de Mello and Jose Antonio Moreira Xexeo. “Cryptographic algorithm identification using machine learning and massive processing”. In: *IEEE Latin America Transactions* 14.11 (2016), pp. 4585–4590.

[4] Cheng Tan, Xiaoyan Deng, and Lijun Zhang. “Identification of block ciphers under cbc mode”. In: *Procedia Computer Science* 131 (2018), pp. 65–71.

[5] SiJie Fan and YaQun Zhao. “Analysis of cryptosystem recognition scheme based on Euclidean distance feature ex- traction in three machine learning classifiers”. In: *Journal of Physics: Conference Series*. Vol. 1314. 1. IOP Publish- ing. 2019, p. 012184.

[6] Aroor Dinesh Dileep and Chellu Chandra Sekhar. “Identifi- cation of block ciphers using support vector machines”. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE. 2006, pp. 2696–2701.

[7] Flávio Luis de Mello and Jose AM Xexeo. “Identifying Encryption Algorithms in ECB and CBC Modes Using Computational Intelligence”. In: *J. Univers. Comput. Sci.* 24.1 (2018), pp. 25–42.

[8] Xinyi Hu and Yaqun Zhao. “Block ciphers classification based on random forest”. In: *Journal of Physics: Conference Series*. Vol. 1168. 3. IOP Publishing. 2019, p. 032015.

[9] Morris J Dworkin. *Sp 800-38a 2001 edition. Recommendation for block cipher modes of operation: Methods and techniques*. 2001.

[10] Xiuli Yu and Kai Shi. “Block ciphers identification scheme based on randomness test”. In: *6th International Workshop on Advanced Algorithms and Control Engineering (IWAACE 2022)*. Vol. 12350. SPIE. 2022, pp. 375–380.

[11] Hua Shen et al. “An efficient aggregation scheme resisting on malicious data mining attacks for smart grid”. In: *Information Sciences* 526 (2020), pp. 289–300.

[12] Burton S Kaliski Jr and Kevin W Kingdon. “Extensions and Revisions to PKCS# 7”. In: *An RSA Laboratories TechnicalNote, Version* 1 (1997).

[13] A. Kahate. *Cryptography and Network Security*. 3rd. NovaDeli: McGraw Hill Education, 2013.

[14] A. Tanenbaum. *Computer Networks*. 5th. Boston: Pearson,2011.

[15] C. P. Pfleeger and S. L. Pfleeger. *Security in Computing*. Boston: Prentice Hall, 2006.

[16] T. Nie, C. Song, and X. Zhi. “Performance Evaluation of DES and Blowfish Algorithms”. In: *International Conference on Biomedical Engineering and Computer Science (ICBECS)*. Wuhan, 2010, pp. 1–4. DOI: 10.1109/ICBECS.2010.5462398.

[17] O. P. Verma et al. “Performance Analysis Of Data Encryption Algorithms”. In: *3rd International Conference on Electronics Computer Technology (ICECT)*. Kanyakumari, 2011, pp. 399–403. DOI: 10.1109/ICECTECH.2011.5942029.

[18] Joan Daemen and Vincent Rijmen. “The Design of Rijndael: AES - The Advanced Encryption Standard”. In: *Springer* 1423 (1997).

[19] Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Berlin: Springer, 2002. DOI: 10.1007/978-3-662-04722-4.

[20] Mitsuru Matsui et al. “Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis”. In: *International Workshop on Fast Software Encryption* (2000), pp. 39–56.

[21] Douglas R. Stinson. *Cryptography: Theory and Practice*. 3rd ed. Chapman & Hall/CRC, 2006.

[22] P Langley. “John, GH: Estimating continuous distributionsin bayesian classifiers”. In: *Proc. Uncertainty in Artificial Intelligence*. 1995.

[23] Mohammad Mohaiminul Islam and Naznin Sultana. “Com- parative study on machine learning algorithms for sentiment classification”. In: *International Journal of Computer Ap- plications* 182.21 (2018), pp. 1–7.

[24] Philip Resnik. “Using information content to evaluate semantic similarity in a taxonomy”. In: *arXiv preprint cmp-lg/9511007* (1995).

[25] Corinna Cortes and Vladimir Vapnik. “Support-vector net-works”. In: *Machine learning* 20 (1995), pp. 273–297.

[26] Leo Breiman. “Random forests”. In: *Machine learning* 45 (2001), pp. 5–32.

[27] Ke Yuan et al. “A Block Cipher Algorithm Identification Scheme Based on Hybrid Random Forest and Logistic Regression Model”. In: *Neural Processing Letters* (2022), pp. 1–19.

[28] Ke Yuan et al. “A block cipher algorithm identification scheme based on hybrid k-nearest neighbor and random forest algorithm”. In: *PeerJ Computer Science* 8 (2022), e1110.

[29] Bruce Schneier and John Kelsey. “Secure audit logs to support computer forensics”. In: *ACM Transactions on Information and System Security (TISSEC)* 2.2 (1999), pp. 159–176.