Machine Learning for Cryptographic Algorithm Identification

F. M. Barbosa, A. R. S. F. Vidal and F. L. de Mello

***Abstract*—This paper aims to study encrypted text files in order to identify their encoding algorithm. Plain texts were encoded with distinct cryptographic algorithms and then some metadata were extracted from these codifications. Afterward, the algorithm identification is obtained by using data mining techniques. Firstly, texts in Portuguese, English and Spanish were encrypted using DES, Blowfish, RSA, and RC4 algorithms. Secondly, the encrypted files were submitted to data mining techniques such as J48, FT, PART, Complement Naive Bayes, and Multilayer Perceptron classifiers. Charts were created using the confusion matrices generated in step two and it was possible to perceive that the percentage of identification for each of the algorithms is greater than a probabilistic bid. There are several scenarios where algorithm identification reaches almost $97,23\%$ of correctness.**

***Keywords*—Cryptographic Algorithm Identification, Data Mining, Machine Intelligence.**

I. Introduction

The theme of this paper is cryptogram analysis in order to identify the cryptographic algorithm used for ciphering. Therefore, it aims to analyze segments from encrypted texts and use this information to identify those algorithms. Even though this test evaluates four cryptographic algorithms, the methodology is generic so that it can be applied to a greater set of algorithms.

The cryptographic algorithms are necessary in order to provide data confidentiality, integrity, authenticity and irreversibility, allowing only the emitter and the receptor of an encrypted message to access the original information content. Today, the cryptographic security depends on the key resistance to attacks and not on the obscurity of the algorithm, that is, the encryption key unknown but the algorithms method are notorious. There are several of such algorithms with different implementations, some are more popular than others are, either because their easiness for implementing or its performance.

Despite the common knowledge of algorithm implementation, the task of breaking the code is neither simple nor brief. First, it is necessary to find out the algorithm used for encoding, and once identified the algorithm, the efforts for obtaining the original information are restricted to attempts of breaking the cipher by using cryptanalysis. Hence, a straightforward cryptanalysis is a huge task. However, there are smaller and complex reduced activities that combined may allow the successful achievement of the task: to determine the cipher size, to retrieve the cipher key, to discover the type of encoding used for cyphering, and retrieve the encryption algorithm.

This work focus on the identification of algorithms used for encoding plain texts by classifying cryptograms trough data mining techniques. The action of finding the key used in such algorithms as well as reversing the encryption is beyond the scope of this article.

II. Related Work

There are a great variety of cryptography algorithms, a sort of procedure responsible for defining data transformations that cannot be easily reversed by unauthorized users. For instance, DES algorithm was developed by the former NIST institute and was widely adopted by industry. Kahate [1] states that this algorithm was the most used for two decades, although its popularity decreased due to its vulnerabilities. Tanenbaum [2] says that the original algorithm is not so secure, but some upgrades can adjust it to be useful. Pfleeger and Pfleeger [3] point out that its security might be achieved by applying successive techniques of substitution and transposition.

The Blowfish algorithm was proposed as an alternative for DES since this was vulnerable to brute force attacks and to others cryptanalysis approaches [4]. Since Blowfish was created to replace DES, some works focus on the comparison among those algorithms. Nie, Song and Zhi [5] provide interesting comparison of speed and energy consumption. Verma, Agarwal, Dafouti and Tyagi [6] demonstrated that the Blowfish is not only faster than DES, AES and Triple DES, but also provides a security enhancement because of its key size. Poonia and Yavad [7] show that some modifications can be made in order to make the algorithm more compact and safer than its original version.

RC4 is a patented algorithm widely used on stream cipher security software such as TLS, SSL and WEP [8]. It is also known as ARC4, since it was never released by RSA, even though its source code was leaked on the Internet [9]. Despite being a simple and efficient algorithm, easily implemented, and five times faster than DES [8], [21], there are several weaknesses that can be exploited [11], [12], [19]. According to Vanhoef and Piessens [10] RC4 should not be used any more.

RSA is the most known asymmetric algorithm [15] and was the first of such algorithms published in literature [16]. Its security relies on the difficulty of factoring very large prime numbers. Coutinho [15] shows that those prime numbers must be wisely chosen, otherwise it is relative simple to break this

F. M. Barbosa, Web Developer with mainframe platform experience, Rio de Janeiro, RJ, Brasil, flaviombarbosa@gmail.com

A. R. S. F. Vidal, Java Programmer with event processing experience, Rio de Janeiro, RJ, Brasil, arthurreimao@hotmail.com

F. L. de Mello (D.Sc.), Assistant Professor at Electronics and Computation Engineering Department from Polytechnic School at Federal University of Rio de Janeiro, Rio de Janeiro, RJ, Brasil, fmello@del.ufrj.br

kind of encryption. Notwithstanding, RSA has been used for encoding and decoding medical images [18] and for a hybrid Bluetooth communication algorithm [17].

Data mining is a process that uses several algorithms in order to retrieve valid patterns from large datasets that can be potentially useful in decision-making process [13], [14], [20]. J48 classifier is an implementation of the classic C4.5 decision tree data mining algorithm, and there are two possible pruning methods [22] to reduce time complexity. The Multilayer Perceptron, on its turn, is a neuron network classifier that is also widely known [26]. It consists of an input layer, intermediate layers and an output layer, where the classificatory training phase is supervised, with backpropagation as a method for minimizing error.

PART is a rules induction method that combines the approaches from C4.5 and RIPPER algorithms without performing a global optimization to produce rules set. The central idea is to create partial decision tree dividing the dataset as in C4.5. Once the partial tree is defined, one rule is obtained from the best leaf node. Gama [24] calls attention to functional trees as an ongoing approach for machine learning and decision models.

FT classifier [24] belongs to an algorithm family, which analyses the differences between decision models. It is similar to several other functional tree algorithms, but the nodes are created according to the samples provided. Additionally, the attributes used in the classification model are incorporated on demand. Besides, the algorithm provides decision lists organized by the set of rules [22].

Naive Bayes is a classifier commonly used as text classifier because its good speed performance, but it also some weaknesses. Rennie et al [25] present two of those weaknesses that influence its performance: 1) the different amount of data for each classes influence the decision weights definition for those categories; 2) the hypothesis of non-overlapping classes. The Complement Naive Bayes is a classifier proposed by Rennie et al [25] which aims to improve Naive Bayes by solving faults associated to misleading trainings.

III. Cryptographic Algorithm Detection

The strategy adopted to detect the algorithm used on a text encryption is to apply data mining algorithms over a set of encrypted files metadata. In order to support this experiment, several plain text files were collected from three different languages. Each file was encrypted with DES, Blowfish, RC4 and RSA algorithms. Then, descriptive metadata was extracted from the cryptograms, that is the sequences bit quantity. Afterward, the data mining procedures were executed by using J48, FT, PART, Complement Naive Bayes and Multilayer Perceptron classifiers. Finally, by using a confusion matrix generated at each mining algorithm execution, it was possible to create an estimative of successful identification of the cryptographic algorithm. Those stages are detailed as follows and illustrated at Fig. 1.

The plain texts used in this experiment encompass three distinct idioms corpora. It was chosen two latin idioms (Portuguese and Spanish) and one anglo-saxon idiom (English).

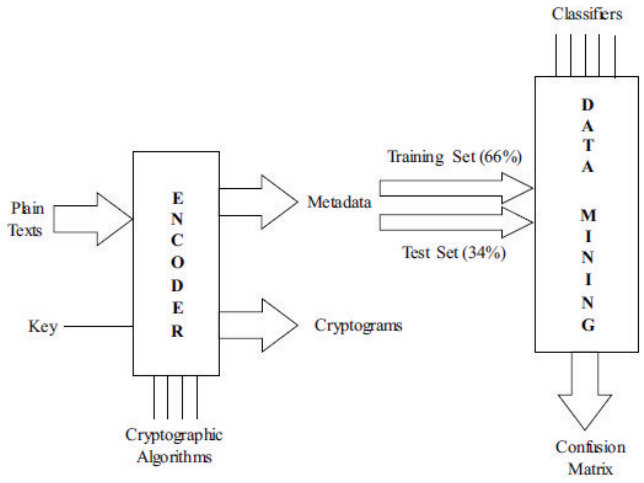

Figure 1. The block diagram of the experiment.

From the point of view of computer compilation, the two former idioms are linguistically more complex but they share some lexical and grammatical similarities. The latter idiom is quite different from the others but it has a simplest structure.

The choice for using those idioms is due to the possibility of comparing the results for different languages and evaluate the sensibility of the cryptographic algorithm classifiers under those circumstances. Each corpus is composed of 200 samples of distinct plain texts, not subject oriented, extracted from newspapers and magazines, without text fragments repetition, and at least with 6.000 characters each.

This study is exploratory in nature and therefore, there is no need for dealing with the most up to date cryptographic algorithms, as a result classical ones were chosen to be evaluated. The criterion for choosing the algorithms is to evaluate the detection behavior for block ciphers, stream ciphers, symmetric key algorithms and public key algorithms. For that reason, the instances of such algorithm classes that were chosen are DES, Blowfish, RSA and RC4.

By the end of the text encryption it is constructed a histogram of each file in order to provide metadata about the cryptograms. The first step for constructing this histogram is to express the number of bits that will define a block, and this block size varies from 4 bits to 16 bits. The bits inside a block correspond to values and then it counts how many values fall into that block.

The automation of text files encryption, for the four cryptographic algorithms, and the histogram construction conditioned to different block sizes was accomplished by using an application developed by Reimão [27]. Hence, three new encrypted files corpora were produced, with a set of corresponding histograms as metadata.

Finally, the metadata was submitted to the classification process containing a set of classifiers. Similar to the process of choosing the cryptographic algorithms, the choice for using classification algorithms instances is based on the viability of employing mining algorithms to the detection of the encryption procedure. By this reason, it was chosen representative algorithms of the classification categories, that is, bayesian

class, functional class, rules based class and decision trees generators class. Therefore, the instances of such classes used at this work are: J48, PART, FT, Complement Naive Bayes and Multilayer Perceptron.

Before performing the tests with the encrypted files corpora, it was necessary to construct the classification model for each classifier. Each encrypted files corpus was segmented into two distinct sets. The first one, containing 66% of the corpus, and was destined to the classification model creation. The second set, containing 34% of the corpus, was submitted to tests. For each individual set from the corpus, there was the same amount of cryptograms encoded with a given encryption algorithm. This feature avoids an algorithm identification enhancement against other algorithms.

IV. Results

The creation of all encrypted files took 42.3 minutes at an i3-2330M CPU @ 2.20 GHz with 4GB RAM memory. The files encoding is a fast procedure but the creation of all histograms, subjected to all possible block sizes, for all files from each idiom, is time costly. The next stage, the data mining stage, is highly dependent on the classifier algorithm. Besides, the execution time from all classifiers increases with the increment of the number of bits of the block, as expected. Moreover, the number of blocks increases as a power of two. These features create a bad environment for processing all combinations of <encrypted file, block size, mining algorithm> in order to create the classification models.

The J48 algorithm took 33.6 minutes of execution time, where 22.5 minutes were spent on constructing the classification model. The FT algorithm spent 7.46 hours executing, from which 2.46 hours were spent on constructing the model. The PART classifier constructed the model in 48.53 minutes from the total execution time of 73.33 minutes. The Complement Naive Bayes is the faster classifier, taking 55 seconds of execution time and just 4 seconds for building the model. At last, the Multilayer Perceptron was the most time consuming, it needed 62.09 hours for constructing the model and 68.91 hours of total execution time. One important notice is that the experiment with the Multilayer Perceptron was not fully accomplished because it was necessary to interrupt its execution. The neural network training is too slow, and thus the time for constructing the model became unfeasible as it increases exponentially. Therefore, it was necessary to limit the block size to 11 bits, that is, blocks from 12 to 16 bits were not evaluated because they are too time expensive. The block with 11 bits size, for instance, took 68.91 hours. Additionally, there were also problems with memory consumption.

The results analysis obtained from the classification process is based on confusion matrix. It aims to get an effective measure of the classification models, since those matrices makes explicit the number of correct classifications versus the number of inferred classifications for each cryptographic algorithm. This means that it was possible to compute the correctness percentage for each classifier applied to each encoding algorithm.

The plots presented in this section describe the classifiers performance for a sample with all three idioms mixed. There is a marginal difference between the performances of the classifiers when using distinct idiom corpus. In fact, the classification got better results for the English corpus, but those results are nor significantly different from the results obtained with the other two corpora. Therefore, it does not seem important to distinguish the idiom of the corpus, not only to define the classification model, but also to be used as test set.

The chart with the results obtained from J48 classifier is presented at Fig. 2. It shows that when using block size of 16 bits the correctness ratio for three algorithms (DES, Blowfish e RSA) is higher, while this classifier combined with the block size criterion is not much sensible for RC4. It is interesting that the smallest block size is a better descriptor because it reaches 61.88% of correctness, while the same block size of 4 bits provides an approximate correctness of 30% for the other classifiers. Moreover, it is observable that the better-identified algorithm under these circumstances is the RSA, with 87.77% of correctness mean value.

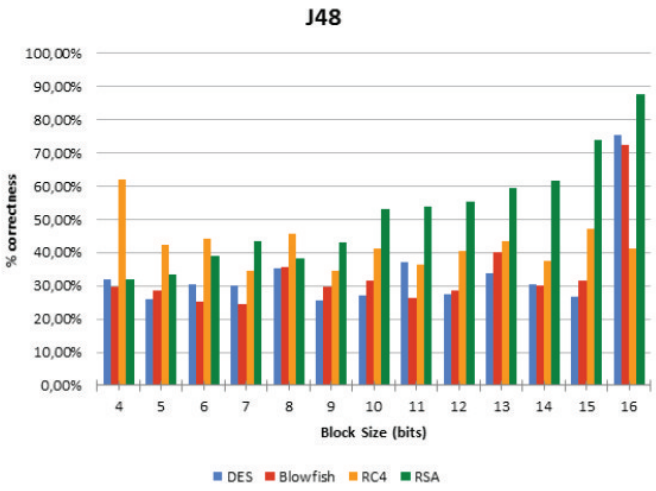

Figure 2. Correctness of J48 classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

Fig. 3 shows the results for the FT classifier. The RC4 encoding is easily recognized by this mining algorithm because the correctness mean value is 66.01% (considering all block sizes). Notice that the 4 bits block size, as it happened to the J48 classifier, is also a good parameter to identify the RC4. However, block with 8 bits provide an even better discriminant. The correctness ratio for DES, Blowfish and RSA reaches its major value with 16 bits block size, as it happened to the J48.

The chart for the PART classifier is presented at Fig. 4 where it is possible to observe a scenario similar to what happened with J48 and PART classifiers. When using the 16 bits block size, the correctness ratio reaches the best results for DES, Blowfish and RSA algorithms. The RC4 algorithm identification is also not so sensible to this classifier. Nevertheless, the usage of PART indicates that the RC4 is again easier classified by using 4 bits block size, as it had already happened with J48 and somewhat with FT. Thus, it seems reasonable that the usage of 4 bits block size can be useful for RC4 identification. The DES and Blowfish identification became a little bit lower

when using 16 bits block size compared to FT, but the RSA identification increased the correctness mean value to 82.45%.

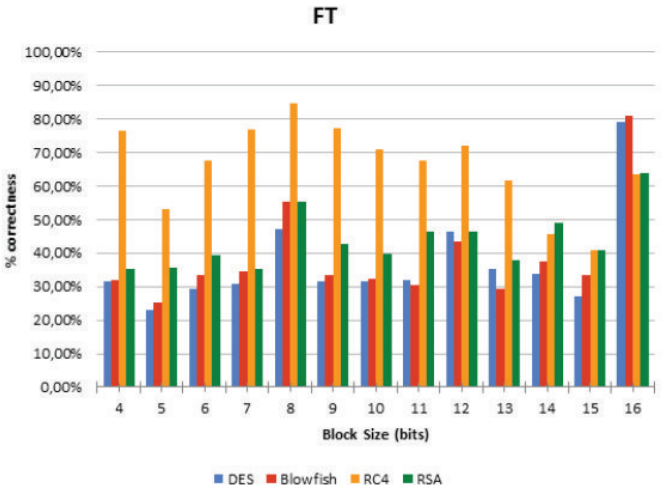

Figure 3. Correctness of FT classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

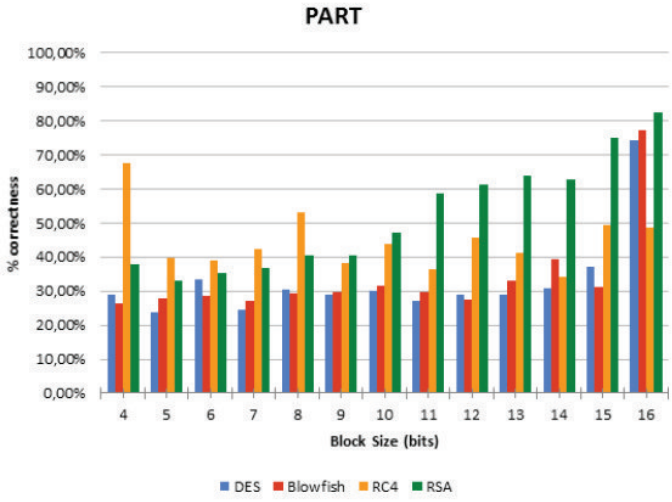

Figure 4. Correctness of PART classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

The chart for Complement Naive Bayes classifier (Fig. 5 shows a completely different scenario. The correctness ratio increased significantly, whatever encoding algorithm is taken for analysis. The Blowfish algorithm is almost fully recognized, and its corresponding correctness mean ratio is 99.54%. The correctness mean value for RSA algorithm increased to 89.36%. DES and RC4 were fully recognized. Additionally, the RC4 algorithm is fully recognized using 8 to 16 bits lock size, in contrast to what had happened with the other classifiers.

At last, the Multilayer Perceptron chart is presented at Fig. 6. Remember that it was not possible to compute the classification model for blocks with 12 bits or more because of technical constraints. The computer platform used in this experiment (Intel i3, 4GB RAM and Windows 10) does not have enough memory to train a neural network with 2^{12} inputs (or more: $2^{13}, 2^{14}, 2^{15}, 2^{16}$) and 4 outputs. This causes a memory fault during this process. Additionally, the time spent to train the neural network was too long. Therefore, the chart from Fig. 6 shows results for 4 to 11 bits block size. The RC4 encoding is better recognized than the others are, and it has a higher correctness mean value for all block sizes. The 11 bits block size is the best parameter for RSA identification and the 8 bits block size is the best option for identifying DES and Blowfish.

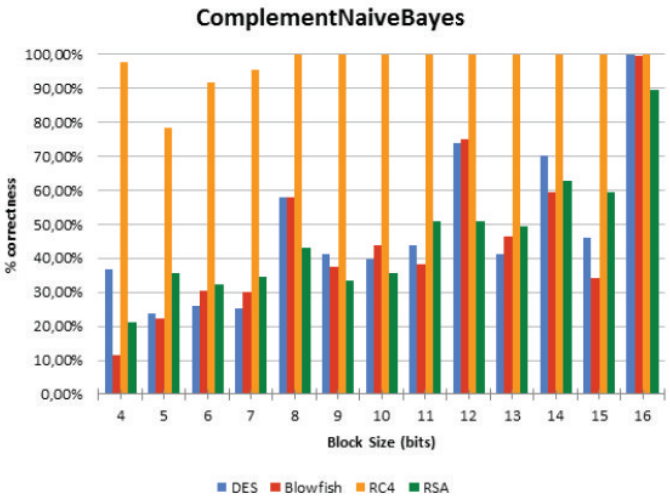

Figure 5. Correctness of Complement Naïve Bayes classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

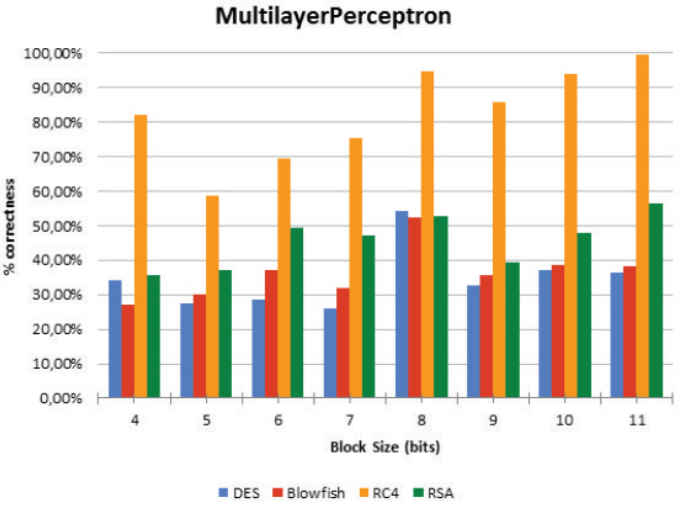

Figure 6. Correctness of Multilayer Perceptron classifier, with block size of 4 to 16 bits, and corpus with tree idioms mixed up.

Taking into consideration that four cryptographic algorithm identification is being studied, the bid for correctly choosing the class of algorithm for a given cryptogram is 25%, with any data analysis and with random selection. For all plots presented at this section it is possible to observe that the correctness mean values for all classifiers are greater than this 25%, even though considering the worst and marginal values of 30% to 33%. However, there are significantly higher

values for correctness ratio, which indicates that the usage of data mining algorithms is useful for encoding algorithm identification.

V. Conclusions

The task of discovering the algorithm used in the process of encoding plain text is not simple. During the cryptanalysis, this knowledge may reduce the efforts for obtaining the original message and thus compromising the information security. Even though this information is not deterministic for breaking the code, this approach of identifying the encoding algorithm can be a useful tool.

This work studied classical cryptographic algorithms identification with the support of machine learning. It showed the viability of classifying cryptograms, according to their encryption algorithm, by using data mining techniques. At this experiment, the random probability for guessing those algorithms is 25%. However, the mean value of correctness obtained here reaches $97,23\%$. In addition, it seems that it is possible to increase this value.

It is important to say that this study was subjected to several constraints. The results obtained here suggest that more efforts must be dedicated to this theme. Despite using a medium size sample space, it was possible to infer the correct cryptographic algorithm with a very good certainty. Therefore, in future works, the number of text samples should be enlarged. It is possible that the successful identification may saturate after using a given number of samples.

Moreover, the size of the blocks should also increased since small blocks do not provided many variation options and thus are not good discriminators. Blocks with 4, 8 and 16 bits size seems to be good parameters. It is reasonable to interested on the performance of block size with 24 and 32 bits. It is also interesting to evaluate if there is another block size that is significant for classification. Besides, it is curious why RC4 is so responsive for 4 bits block size, although this encryption algorithm is no longer relevant.

Furthermore, the set of idioms did not influence the classification process, but it is not guaranteed that those idioms are sufficiently different. The inclusion of other idioms with distinct alphabets and grammars, such as Bulgarian, Russian, Swedish, Chinese, and Arab, for instance, may provide the necessary variations for idiom sensitiveness.

Additional cryptographic algorithms are also a good improvement, so that the results obtained can be used in real applications. Different instances of encoders with Electronic Code Book (ECB), Cypher Block Chaining (CBC), Cypher Feedback (CFB) and Output Feedback (OFB) are also necessary to be studied.

The Complement Naive Bayes algorithm seems to be the best classifier, but this ranking can be changed when the number of different cryptographic algorithms increase. Likewise, more mining algorithms may be tested. One of those algorithms is the Weightless Neural Network, which can substitute the Multilayer Perceptron for fast training and classification.

Acknowledgments

The Federal University of Rio de Janeiro (UFRJ) Coppetec Foundation supported this research, under project Poli-19.257, which was carried out at the Machine Intelligence Laboratory of UFRJ.

References

[1] Kahate, A. *Cryptography and Network Security*, 3rd ed, Nova Deli, McGraw Hill Education, 2013.
[2] Tanenbaum, A. *Computer Network*, 5th edition, Boston, Pearson, 2011.
[3] Pfleeger, C. P. and Pfleeger, S. L. *Security in Computing*, Boston, Prentice Hall, 2006.
[4] Schneier, B. *Fast Software Encryption*, Cambridge Security Workshop Proceedings, pp. 191-204, 1994.
[5] [10] Nie, T., Song, C., Zhi, X. *Performance Evaluation of DES and Blowfish Algorithms*, International Conference on Biomedical Engineering and Computer Science (ICBECS), pp. 1-4, Wuhan, 2010.
[6] Verma, O. P., Agarwal, R., Dafouti, D., Tyagi, S. *Performance Analysis Of Data Encryption Algorithms*, 3rd International Conference on Electronics Computer Technology (ICECT), pp. 399-403, Kanyakumari, 2011.
[7] Poonia, V., Yadav, N. S. *Analysis of modified Blowfish Algorithm in different cases with various parameters*, International Conference on Advanced Computing and Communication Systems, pp. 1-5, Coimbatore, 2015.
[8] Hammood, M. M., Yoshigoe, K., Sagheer, A. M. *RC4-2S: RC4 Stream Cipher with Two State Tables*, Information Technology Convergence, v. 253, pp. 13-20, 2013.
[9] Paul, G., Maitra, S. *RC4 Stream Cypher and Its Variants*. Boston, CRC Press, 2012.
[10] Vanhoef, M., Piessens, F. *All Your Biases Belong To Us: Breaking RC4 in WPA-TKIP and TLS*, Proceedings of the 24th USENIX Conference on Security Symposium, pp. 12-14, Washington, 2015.
[11] Fluhrer, S., Mantin, I., Shamir, A. *Weakness in the Key Scheduling Algorithm of RC4*, Selected Areas of Cryptography, v. 2259, pp. 1-24, 2001.
[12] Mantin, I., Shamir, A. *A Pratical Attack on Broadcast RC4*, Fast Software Encryption, v. 2355, pp. 152-164, 2002.
[13] Navega, S., *Princípios Essenciais do Data Mining*, Anais de Infoimagem, Cenadem, 2002.
[14] Han, J., Kamber, M., Pei, J. *Data Mining Concepts and Techniques*, 3rd edition, Morgan Kaufmann, Waltham, 2011.
[15] , Coutinho, C. S. *Números Inteiros e Criptografia RSA*, IMPA, Rio de Janeiro, 2003.
[16] Das, A., Madhavan, C. E. V. *Public-key Cryptography Theory and Practice*, Deli, Pearson, 2009.
[17] Ren, W., Miao, Z. *A Hybrid Algorithm Based on DES and RSA in Bluetooth Comunication*, Second International Conference on Modeling, Simulation and Visualization Methods (WMSVM), pp. 221-225, Sanya, 2010.
[18] Anane, N., Anane, M., Bessalah, H., Issad, M., Messaoudi, K. *RSA Based Encryption Decryption of Medical Images*, 7th International Multi-Conference on Systems Signals and Devices (SSD), pp. 1-4, 2010.
[19] Goutam, P., Subhamoy, M. *RC4 State Information at Any Stage Reveals the Secret Key*, IACR Cryptology ePrint Archive, 2007.
[20] Witten, I. H., Frank, E., Hall, M. A. *Data Mining Practical Machine Learning Tools and Techniques*, 3rd edition, Morgan Kaufmann, Burlington, 2011.
[21] Gupta, S., Chattopadhyay, A., Sinha, K., Maitra, S., Sinha B. *High-performance hardware implementation for RC4 stream cipher*, IEEE Transaction Computers, v. 62(4), pp. 730-743, 2013.
[22] Mohamed, W. N. H. W., Sallen, M. N. M., Omar, A. H. *A Comparative Study of Reduced Error Pruning Method in Decision Tree Algorithms*, IEEE International Conference of Control System, Computing and Engineering, Penang, pp. 23-25, 2012.
[23] Frank, E., Witten, I. *Generating Accurate Rule Sets Without Global Optimization*, Proceedings of the Fifteenth International Conference on Machine Learning, pp. 144-151, São Francisco, 1998.
[24] Gama, J. *Functional Trees*, Machine Learning, v. 55(3), pp. 219-250, 2004.
[25] Rennie, J. D. M., Shih, L., Teevan, J., Karger, D. R. *Tackling the Poor Assumpltions of Naive Bayes Text Classifiers* Proceedins of the Twentieth International Conference on Machine Learning, Whasington DC, 2003.

[26] Silva, L. N. C. *Análise e Síntese de Estratégias de Aprendizado Para Redes Neurais Artificiais* Projeto de Mestrado, Universidade Estadual de Campinas, Setembro de 1998.
[27] Reimão, A. S. F. V. *Análise de blocos de arquivos criptografados para obtenção do algoritmo*, Projeto de Graduação, Universidade Federal do Rio de Janeiro, Fevereiro 2015.

Flávio Mendonça Barbosa did his MBA on computer and systems engineering at Federal University of Rio de Janeiro - UFRJ (2016) and undergraduation on computer science at Federal University of Rio de Janeiro - UFRJ (2012). Worked on mainframe platform for six years and has been developing web apps with Java as the backend platform since 2012.

Arthur Reimão Santos Figueiredo Vidal did his undergraduation on electronics engineer at Federal University of Rio de Janeiro - UFRJ (2014). Worked with event processing for a year and a half and is currently studying for a public sector job.

Flávio Luis de Mello did his DSc. on theory of computation and image processing at the Federal University of Rio de Janeiro - UFRJ (2006), MSc. on computer graphics at Federal University of Rio de Janeiro - UFRJ (2003), under graduation on systems engineering at Military Engineering Institute - IME (1998). Developed command and control systems and implemented military messages interchange applications during twelve years as Brazilian Army officer. Responsible for developing software applications based on theorem proving, knowledge base systems and knowledge representation from Mentor Group. Associate Professor at the Electronics and Computing Department (DEL) of Polytechnic School (Poli) at Federal University of Rio de Janeiro (UFRJ) since 2007.