Research on Cryptographic Algorithm Recognition Based on Behavior Analysis

Fei Yan[1,2]，Yunlong Xing[1,2]，Shiwei Zhang[2]，Zhihan Yue[2] and Yamin Zheng[2]

(1. Key Laboratory of Aerospace Information Security and Trusted Computing of Ministry of Education, Wuhan 430072, China;

2. School of Computer, Wuhan Univ., Wuhan 430072, China)[1]

Email: yanfei@whu.edu.cn, yunlong.xing@whu.edu.cn

Abstract. Due to the abuse of cryptography technology and the difficulty to break encryption algorithm, ransomware has a huge threat to cyberspace. So how to detect the cryptographic algorithm in the recognition program plays an important role in the protection of information security. However, existing cryptographic algorithm identification and analysis technology has the disadvantages of low recognition efficiency, single analysis strategy, and they cannot identify program variants effectively. In view of these problems, this paper presents a cryptographic algorithm based on behavior analysis. Based on the behavior analysis, combined with the static structure and dynamic statistical characteristics of the key data, the subroutine of the target program is gradually screened, and the execution logic of the subroutine is analyzed. Finally, the cryptographic algorithm in the binary code of the program is obtained. Compared with the traditional signature-based technology, our technology has a better recognition rate with less resource occupation. What's more, this technology can identify the program variants accurately, so it has a good application prospects.

Key words: Cryptographic algorithm identification, Behavior analysis, Signature recognition

0 Introduction

With the popularity of the Internet, computer application technology and network communication technology develop rapidly, especially in the process of information exchange, cryptography plays an important role in ensuring information security. However, if a high-security cryptographic algorithm is exploited by malicious programs, it poses a huge threat to the critical data and file systems of the average user. So, it is important in the identification and analysis of binary code in the password algorithm for malicious code detection and password security analysis.

The traditional cryptographic algorithm identification is mainly based on the idea of signature matching [1], by analyzing the existing cryptographic algorithm and extracting the corresponding eigenvalue, the feature library is constructed. When the target program is detected, the characteristic

This work is sponsored by National Science Foundation of China (No. 61272452), National High-tech R&D Program of China (863 Program) 2015AA016002, and National Basic Research Program of China (973 Program) 2014CB340601.

value of the program is matched with the library feature. If matching succeeds, the program contains built-in cryptographic algorithm. Many anti-virus software is based on the construction of the signature library. But there are several obvious flaws in the method of signature matching: 1) the identification is not in time, only after obtaining a program sample characteristics, we can detect the program; 2) rely on the upgrading of the signature database and consume system resources seriously; 3) cannot detect new malware. Although the development of technology, the detection has more rich characteristics, and classification technology is more accurate, but if the statistical code is too rigid, the malicious program is easy to avoid the detection of signatures [2] [3] [4] [5] [6].

In addition to signature matching, there are some new cryptographic algorithm recognition techniques, for example, the detection of cryptographic algorithm based on the cipher text randomness [7] and the cipher text statistics detection [8]. We can predict the cryptographic algorithms of the program to be tested by statistical correlation of cipher text, but this detection technique is complicated and cannot recognize the public key cryptography algorithm effectively.

In view of these problems, this paper presents a new technology for cryptographic algorithm recognition based on behavior analysis. Based on the behavior analysis and the static structure and dynamic statistical characteristics of the key data, the subroutine of the target program is gradually screened and positioned after the encryption subroutine to analyze the execution logic. Then, according to the different characteristics of each type of cryptographic algorithm, we can identify the code in the binary code of the program accurately.

The main contributions of this paper are as follows:

1) Based on behavior analysis, and focusing on program execution results and algorithmic logic, this technology has a high efficiency for the program using traditional cryptographic algorithmic;

2) Combined with the binary code of the program static feature matching and compilation level instruction dynamic piling comprehensive analysis, false negative rate and false positive rate are low and the accuracy is very high;

3) Provide new ideas for new cryptographic algorithms and provide technical support for malware protection.

1 Related Technology

In the process of realizing the system, the DynamoRIO dynamic binary jacking platform is used to extract the index of metric of training data set and the naive Bayesian classifier is used to train data for coming into been probability model. The following two techniques will be introduced briefly and respectively:

1.1 DynamoRIO Dynamic Binary Jacking Platform

DynamoRIO is the simulation software at the process level working between the application layer and the operating system, whose working principle is been showing below:

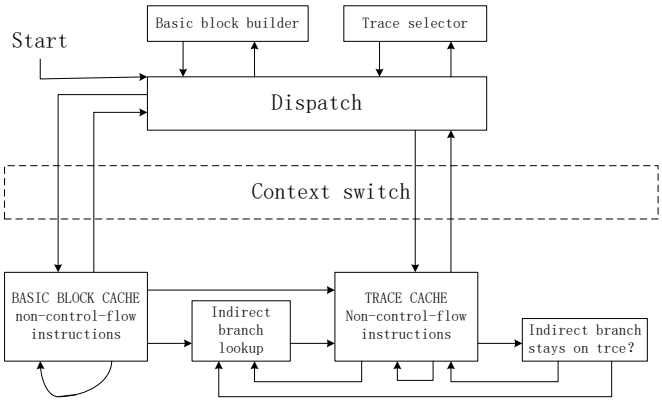

Fig. 1. The working principle of each component

As shown in Figure 1, when DynamoRIO platform is debugging the program plug, the program will call the core components of the platform first. Then the code will be divided into basic blocks by built-in basic block builder and stored in the basic block cache. At the same time, non-control flow instruction in the cache will be running in sequential way until it encounters a control flow instruction. The tracking selection module then counts the call relationship between the basic blocks based on the control flow instruction and determines whether continuing the trace according to the context or not.

To realize the efficient of instrumentation and debugging, DynamoRIO uses many kinds of optimization strategy such as code caching, linking, tracking builds and transparent use of resources. The code cache uses the local principle to use the basic block as a unit for the local use of the code cache processing to improve the efficiency of instruction read. The code cache uses the local principle to use the basic block as a unit for the local use of the code cache processing to improve the efficiency of instruction read. If target instructions already exist in code cache, and pointed by the direct jump instruction, DynamoRIO can jump directly to the code cache target instructions, to avoid context switching overhead. And in the process of debugging, DynamoRIO tries to be transparent and does not affect the use of the program itself.

1.2 Naive Bayesian Classifier

Native Bayesian classifier is a very simple classification algorithm, whose ideological basis is as follows. As for the items to be sorted, native Bayesian classifier can calculate the probability of occurrence of each category under the condition of this occurrence of the items and match with maximum probability value [10]. Its formal definition is as follows:

Step 1： Set x = {a1, a2, a3……am}as one item to be sorted ， every a as one feature attribute of x;

Step 2： Set category collection C = {y1, y2, y3……yn};

Step 3： Calculate P(y1|x), P(y2|x), ……, P(yn|x);

Step 4： If P(yk|x) = max{P(y1|x), P(y2|x),……, P(yn|x)}, x∈yk.

The key of the definition is the calculation of every conditional probability in step 3. First, find a collection of known categories to be classified, then add up conditional probability of every feature attribute to this category. That is to say P(a1|y1), P(a2|y1),…, P(am|y1)； P(a1|y2), P(a2|y2),…, P(am|y2)； …； P(a1|yn), P(a2|yn),…, P(am|yn). If every feature attribute condition is independent, we can get conclusion according to Bayesian formula which is:

$$P(yi|x)=(P(x|yi)P(yi)) \ / \ (P(x)) \quad (1)$$

2 System Design and Analysis

This paper is based on the behavior analysis of the program to identify the cryptographic algorithm. The technology is based on the behavior analysis, through the subroutine screening in the target program, and gradually positioning to the encryption subroutine, and then analyze its execution logic. With the helping of static structure matching and dynamic statistical features, you can accurately identify the program binary code in the cryptographic algorithm and the algorithm accurate positioning. The general idea of cryptographic algorithm recognition is as follows:

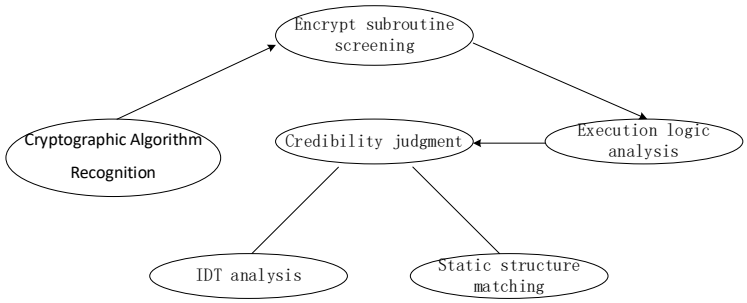

Fig. 2. Design method of system design

In order to realize the automatic identification of the system, we define the filtering rule of the encryption subroutine：1) encryption subroutine is usually located in the subroutine call tree leaf node; 2) the number of basic block in encryption subroutine will not exceed 20; 3) the relative proportion of instruction mov in encryption subroutine is usual 40%-60% [9]. The execution logic analysis is mainly for the hash function, packet encryption algorithm, public key encryption algorithm and stream cipher. By extracting chain value initialization, cyclic shift and S box initialization, modular exponentiation and stream key generation logic, we can match with the general encryption algorithm. IDT (import directory table) analysis is mainly for the process of implementation of reference to the dynamic link library and analyses whether to call suspicious library functions, such as registry operations and reference encryption function library. Finally, the static structure matching is applied to the initialization chain value in hash function and the matching of the static structure such as the S-box and the P-box in packet encryption function. Through the execution logic analysis and help with the IDT analysis and static structure matching, we can make an accurate decision in the credibility of the program. What's more, the system introduces DynamoRIO dynamic binary analysis platform and naive Bayesian algorithm, the system frame is as follows:

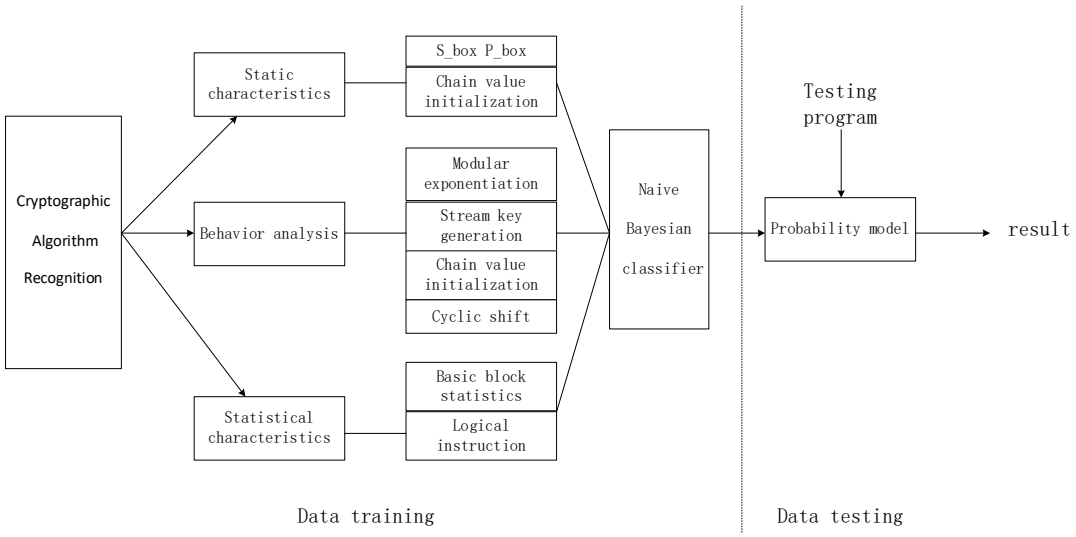

Fig. 3. System implement overview

As shown above, the system is based on naive Bayesian algorithm, which is simple and efficient, and trains a small amount of data to draw the necessary parameters. By extracting the static eigenvalues of the program, the DynamoRIO dynamic binary analysis platform is used to extract the statistical features, and the training elements are introduced into the Bayesian classifier in conjunction with the logic analysis, thus the probability model is obtained. Therefore, the system implementation mainly includes the extraction of meta-meta-features, data training and data testing three stages.

2.1 Extract Metric Features

In the extraction meta-feature stage, many programs are mainly divided into two sets according to whether or not they contain the cryptographic function: the cryptographic assembly and the ordinary assembly. Then, DynamoRIO dynamic binary jacking platform is used to simulate the operation of each program, and calculates the probability relation between the program and the correlation quantity.

Extract static features. The static characteristics of this part in the extraction program mainly include S box and some static matrix structure. S box is widely used in block cipher, completing the confusion of cryptographic algorithms, and maintaining the relative stability in the process of implementation. For example, in AES encryption algorithm, S and S^{-1} are both 16 * 16 matrix, completing one mapping 8bit input to 8bit output; the DES function uses eight S-boxes, and each S-box is converted into the 6bit input to 4bit output. S box construction is generally in front of the implementation of the process and exists with static structure in the program for the internal table operation [11].

The static matrix is another static structure in the implementation of the encryption algorithm, which mainly completes the diffusion of the cryptographic algorithm. Diffusion is used to make the relationship between plaintext and cipher text acting as complex as possible. Any small changes in the plaintext will make the cipher text very different, and the realization of the diffusion is mainly based on the matrix multiplication. For example, in the AES encryption algorithm, the row shift is a 4 * 4 matrix for byte-to-byte permutations that provide the diffusivity of the algorithm. Column confusion is also used to adjust the relative position of the elements of the matrix transformation to improve the security of the algorithm. During the implementation, several key constants of the column confusion matrix are: 0x02, 0x01, 0x01,

0x03.

The static feature extraction flow chart is as follows:

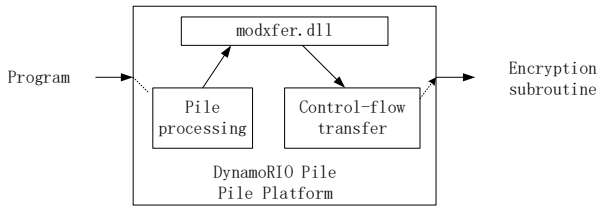

Fig. 4. Extract static features

As shown in Figure 4, in order to identify whether the program contains S-box and static matrix and other key static structure, we first use the DynamoRIO dynamic stakes platform to run the test program, and then use the platform to compile the dynamic link library modxfer.dll on the target program Pile processing, reporting the flow of control flow between the modules, through the control flow call relationship to the encryption subroutine and its program entry, use the hexadecimal editor to open the program, jump to the encryption subroutine to static structure Of the keyword search, according to the search results and a small amount of experience we can determine whether the program contains S boxes and static matrix and other static structure.

Extract Dynamic Characteristics. The dynamic characteristic, a kind of statistical characteristic, is the obvious difference shown in the cryptographic program and the ordinary program in the process of executing the program. The dynamic extraction of this part mainly includes the relative proportion of the mov instruction of the subroutine during the execution of the program and the multiple cyclic encryption mode.

The mov instruction carries out the transfer function in the assembly level code, transfers the data between the CPU internal registers, transfers the immediate data to the general register inside the CPU, the data transfer between the register and the external memory, and the immediate value to the storage unit. The In the cryptographic algorithm subroutine, the mov instruction is usually used to complete the lookup operation of static structure such as S box, and the mov instruction is used to complete the temporary transmission of plaintext and cipher text data. Therefore, the relative instruction ratio of the mov instruction in the subroutine General subroutine is much higher.

The loop feature mainly completes the generation of block key in the block cipher algorithm, the iterative encryption, and optimizing the encryption efficiency of modal square residual algorithm in the public key cryptography. In the high-security encryption algorithm, the round robin plays a key role in the RC5 algorithm, the round robin is set to 18-20 to resist the differential analysis. In the DES algorithm, the subkey generates 16 cycles of iteration, and the key pair data is encrypted 16 times.

The dynamic feature extraction flow chart is as follows:

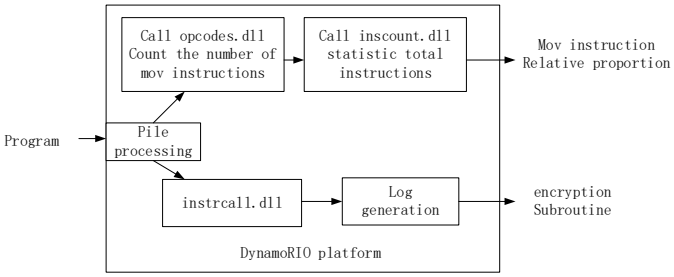

Fig. 5. Extract dynamic characteristics

As shown in Figure 5, in order to extract the mov instruction relative proportion in the process of executing the program, in the implementation of DynamoRIO pile under the premise of the use of inscount.dll dynamic technology all the number of instructions and op codes.dll dynamic implementation of the decomposition of different instructions Code, and report the number, and then the number of instructions to move the number of instruction instructions, the mov instruction in the subroutine in the relative proportion. Through a large number of statistical data analysis, cryptographic algorithm subroutine mov instruction relative proportion should be more than 40%. The extraction of the loop pattern is based on the instrcalls.dll statistics direct call, indirect call and return the log file generated by the instruction to determine the number of iterations of the key function of the encrypted subroutine. According to the statistical results, the number of cycles of the subroutine must be more than 16 times to satisfy the metric measurement requirement.

Behavioral analysis. The behavior analysis of the cryptographic program mainly focuses on the four main encryption algorithms: hash function, packet encryption algorithm, public key encryption algorithm and stream cipher. The hash algorithm is used to initialize the corresponding chain value. The packet encryption algorithm corresponds to the operation of the S-box and the P-box and the cyclic shift. The public key algorithm corresponds to the call of the large tree and the modular exponent operation. The stream cipher algorithm mainly analyzes the generation of the stream key.

Hash functions typically include an initialization module and a summary value generation module when the code is implemented. The initial memory unit requested in the initialization module plays the role of saving the initialized chain value, the intermediate block hash result, and the last hash value. In the disassembly result, the initial chain value is stored in the continuous memory unit (imm is the initial chain value) using successive mov imm, imm instructions, and the continuous memory unit uses add reg1, reg2; mov mem, reg1 or xor Mem, reg instruction form, the last packet processing is completed, the continuous memory unit to store the input data hash value.

In order to implement the "confusion" and "diffusion" of the plaintext data, the block cipher is usually used in static data such as S-boxes and P-boxes in feistel, SP and other network structures and stored in a binary executable file in a binary executable file in the data segment. In the disassembly results, the block password is usually used mov reg1, dword_addr [reg2] instruction on the P box, S box and other static array access.

The public key cryptography usually uses large primes in the implementation process. These large primes are usually generated by the random number generator. In the disassembly result, the RSA algorithm will show more obvious exponentiation operation. To optimize the modular exponentiation efficiency, the algorithm usually uses modulo residual implementation.

RC4 is the most widely used stream cipher, the algorithm generates a byte pseudo-random key stream, which is mixed with the plaintext operation to achieve the purpose of encryption, decryption, the generated pseudo-random flow and cipher text XOR operation to restore the plaintext.

2.2 Data Training

In the stage of data training, the probability statistic model is obtained based on the correspondence between each program and the five metric elements, the relative proportion of the cryptographic program and the common program.

The structure of the stage is as follows:

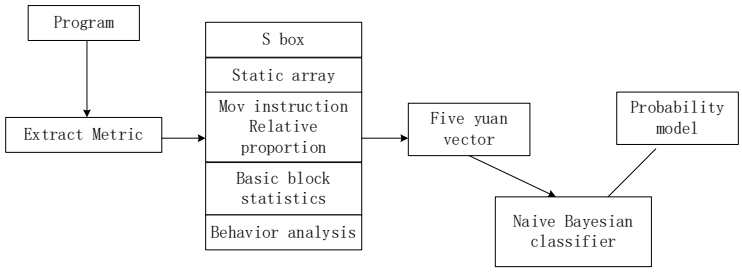

Fig. 6. Data training structure

As shown in Figure 6, five metrics referred to as 5 yuan S_N -S boxes, static matrix- N_{ST}, mov instruction accounting for N mov, cyclic mode N cyc, behavior analysis- N sh, for each of the programs there are (N_S, N_{ST}, N mov, N cyc, N sh) five yuan vectors, each vector element value is 0 or 1.

As can be seen from the equation (1) Bayes' theorem, first we divide all the programs in the training data set into two categories according to whether the program is a cryptographic program, wherein the assembly referred to as a code C 1, referred to as a common assembly C 2. when the training data, all five of first element vector is introduced into all the program code Bayesian classifier, then count the number of each vector element value and class values that satisfy this condition, for example, the S-box train, first Count the number of cryptographic procedures, and then calculate the number of S boxes to meet the characteristics of the two can be compared to the S box can be obtained in the password program probability, the other meta-like solution.

Respectively, in the cryptographic program in the probability of the emergence of P1, P2, P3, P4 and P5, then in the password program, the proportion of each metric is the proportion of:

$$Qi = Pi / (P1 + P2 + P3 + P4 + P5) \quad (2)$$

The “i” above is in the range of 1 to 5, and the proportion of the probability of each meta-element in the common program can be solved by the same token, and the proportion of each metric element in the cipher program and the common program is recorded. The probability matching model of the cryptographic program can be obtained.

2.3 Data Testing

In the data testing phase, the probability model of data training is used to match the metric of the test program, to judge whether the program contains the cryptographic algorithm.

As can be seen from (2), the cryptographic program and the common proportion of each program in the proportion of the proportion of the data in the test phase, for each test program, first in accordance with the provisions of Section 2.1 to extract the characteristics of the five elements of the program (0, 1, 1, 1, 1), and multiply the vector by multiplying the vector by the inverse of the vector The probability of matching the program to the category is as follows:

$$X = (0, 1, 1, 1, 1) * (Q1, Q2, Q3, Q4, Q5)^{-1} \quad (3)$$

To recognize a cryptographic algorithm, we only need to compare X with Y, the expected value. If $X \geq Y$, the program is more likely to be a cryptographic program, and vice versa. Besides, because the password program and the ordinary procedure are opposite events, so we only need to judge one of these two cases.

3 Experiments and Assessment

In this essay, experimental environment is equipped with Intel i5-5200U CPU、8GB RAM、128GB SSD + 512GB HDD、Windows 7 SP1 and DynamoRIO dynamic Pile Platform. Experiments and Assessment in this chapter contain three main stages: data collection, function assessment and effect assessment.

3.1 Data Collection

Collected data in this essay contains 563 programs which include 506 password programs and 57 normal programs and password programs accounted for 90% of whole programs. Password programs are mainly collected from famous on-line virus analysis websit-virustotal and academic support provided by AnTian company. All of code programs are assessed by virustotal and every program has only one category. Data categories are showed in a form below:

Table 1. Collect data classification

	Cryptography	Normal	Total
Training	461	52	513
Testing	45	5	50

As No.1 form demonstrates, all of programs are divided into two parts: training set and testing set. Training set contains 513 programs, including 461 password programs and 52 normal programs. Two kinds of programs belong to different categories. Testing set contains 50 programs, including 45 password programs and 5 normal programs. This two kinds of programs are not distinguished.

3.2 Function Assessment

The aim of function assessment is to test that whether cryptographic algorithm recognition technology can distinguish the password programs from programs or not. The main body of the validation here is whether the naive Bayesian algorithm can generate the probability matching model according to the input training data set and whether the test data set can be detected and identified according to the probability model.

Production of probabilistic models. Production of probabilistic models needs to extract 5 metrics of static characteristics dynamic characteristics and behavioral characteristics and every 5 metric of each programs will produce 0/1 five yuan vector.

First, testing set's static characteristics which is the steady data structure when programs are processing should be extracted. Therefore, once encryption subroutine inside can be located, we can find whether the search subroutine contains the static structure through the string. In section 2.1.1., we know that DynamoRIO binary digit analysis platform can identify programs control flow transfer through the transfer of program dynamic pile and modxfer.dll. and that help us find the entrance of the subroutine. The statistics for this section are as follows:

Table 2. Static feature information collection

	Sbox	**Static array**	**Total**
Cryptography	273	150	423
Normal	3	9	12

It can be seen from Table 2 that the probability that the two static features of the S-box and the static

matrix exist in the cryptographic program is much larger than that of the ordinary program.

The dynamic characteristics include the relative proportion of the mov instruction and the loop pattern. As can be seen from Section 2.1.2, the relative proportion of the mov instruction is to call the opcodes.dll and inscount.dll statistics subroutines in the program via the DynamoRIO platform. The number of instructions and the total number of instructions are compared with 40% -60%. For comparison, you can determine whether the indicator is met. Similar to the static structure, the search of the loop pattern is also done by locating it into the subroutine. From analysis, the part of the statistical results shown in the following table:

Table 3. Dynamic feature information collection

	Mov	Cyclic shift	Total
Cryptography	403	125	528
Normal	10	23	33

As can be seen from Table 3, the general program may also contain more mov instructions and multiple loop structure, but compared to the password program is still less.

The extraction of behavioral characteristics is relatively simple, only need to locate the encryption subroutine, and then analyze its execution logic to determine whether it is modular exponentiation, stream key generation, cyclic shift or chain value initialization of one of them. The results are as follows:

Table 4. Behavioral feature information collection

	Behavioral characteristics	Total
Cryptography	453	453
Normal	5	5

As can be seen from the table, almost all of the password procedures are in line with the metrics.

Data Testing. According to the three tables in 3.2.1, we can get the frequency of the training data and the frequency of the common program. The frequency is approximately equal to the probability, and the following formula can be obtained by combining the formula (2):

Table 5. Cryptographic algorithm probabilistic mode

i=	1	2	3	4	5
Pi	59.2	32.5	87.4	27.1	98.2
Qi	29.3	10.7	28.7	9.0	32.3

From Table 5, we can get the probability model of the cryptographic function, which has the statistical probability and the relative proportion of each metric.

For the test data set, extract the meta-generated 0/1 five-dimensional vector, according to Table 4 data and calculation formula (3), take Y is 65%, that can determine whether the program is a password program.

3.3 Performance Evaluation

In order to compare the effectiveness of the technology, all the programs in the test set were tested and analyzed separately using the technique and the signature detection technique, respectively, the false

alarm rate, the false negative rate and the accuracy rate. The results are as follows:

Table 6. Experimental comparison

	False positives	True negatives	Accuracy
Signature	10%	18%	86%
Behavior	4%	2%	97%

It can be seen from Table 6 that the algorithm is superior to the algorithm based on the signature matching algorithm in terms of false positive rate, false negative rate and accuracy rate, and the technology can identify unknown cryptographic procedures. Therefore, Naive Bayesian classifier cryptographic algorithm identification research technology has obvious practical value.

4 Conclusion

Based on the behavior analysis, combined with the key static structure and dynamic statistical characteristics, the subroutine of the target program is gradually screened, and the execution logic of the subroutine is analyzed. Finally, the cryptographic algorithm in the binary code of the program is obtained. In this process, DynamoRIO dynamic insertion technique is introduced to simulate the running, synchronous tracking and naive Bayesian classifier probability model. Whenever the cryptographic algorithm in the test program is used, the relative ratio of the S-box, the static data, the mov instruction in the subroutine, the cyclic shift operation and the behavioral characteristics are extracted, and then matched with the probability model whether the program contains a cryptographic algorithm inside. Compared with the traditional signature-based recognition technology, the technology occupies less resources, higher recognition rate, but also accurately identify the program variants.

References:

[1] Traynor P, Chien M, Weaver S, et al. Noninvasive methods for host certification. ACM Transactions on Information and System Security (TISSEC), 2008, 11(3): 16.

[2] Chen X, Andersen J, Mao Z M, et al. Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware in Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on. IEEE, 2008: 177-186.

[3] Maiorca D, Corona I, Giacinto G. Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013: 119-130.

[4] Jana S, Shmatikov V. Abusing file processing in malware detectors for fun and profit in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012: 80-94.

[5] Marpaung J A P, Sain M, Lee H J. Survey on malware evasion techniques: State of the art and challenges in Advanced Communication Technology (ICACT), 2012 14th International Conference on. IEEE, 2012: 744-749.

[6] Ugarte-Pedrero X, Balzarotti D, Santos I, et al. SoK: deep packer inspection: a longitudinal study of the complexity of run-time packers in Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, 2015: 659-673.

[7] Wu Yang, Wang Tao, Xing Meng, et al. Recognition Scheme of Block Cipher Algorithm Based on Categorical Randomness Metrics Distribution. Journal of Communications, 2015, 36(4): 147-155.

[8] Wu Yang, Wang Tao, Li Jindong. A New Method of Statistical Detection for Cryptography of Block Cipher Algorithm. Journal of Ordnance Engineering College, 2015, 27(3): 58-64.

[9] Li Jizhong. Research on key technology of cryptographic algorithm identification and analysis. The PLA Information Engineering University, 2014.
[10] Li Jizhong, Jiang Liehui, Yin Qing, et al. Recognition Algorithm of Cryptographic Algorithms Based on Bayes Decision. Computer Engineering, 2008, 34 (20): 159-160.
[11] Liu T M, Jiang L, He H, et al. Researching on cryptographic algorithm recognition based on static characteristic-code. Security Technology, 2009: 140-147.
[12] de Mello F L, Xexeo J A M. Cryptographic Algorithm Identification Using Machine Learning and Massive Processing. IEEE Latin America Transactions, 2016, 14(11): 4585-4590.